

Timo Kopeli

**KVM VIRTUALISOINTI KOMENTORIVIN JA PROXMOX VE:N KAUTTA**

Opinnäytetyö  
Kajaanin ammattikorkeakoulu  
Luonnontieteiden ala  
Tietojenkäsittelyn koulutusohjelma  
Kevät 2014



Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Kopeli, Timo	
Työn nimi KVM virtualisointi komentorivin ja Proxmox VE:n kautta	
Vaihtoehtoiset ammattiopinnot Järjestelmänylläpito	Toimeksiantaja  Ohjaaja(t) Partanen, Timo
Aika Kevät 2014	Sivumäärä ja liitteet 50
<p>Kernel-based Virtual Machine eli KVM on kasvava kilpailija kaupallisille virtualisointialustoille. Se perustuu avoimeen lähdekoodiin ja on integroitu Linux-kerneliin, minkä vuoksi se on ilmainen. Ilmaisuus houkuttelee yrityksiä ja muita tahoja, joilla virtualisoinnista koituvat lisenssikustannukset ovat suuri osa IT-budjettia. Opinnäytetyön tavoitteena oli selvittää KVM:n tämänhetkistä tilannetta ja tekniikoita sekä sitä, kuinka sen avulla voidaan pystyttää palvelinvirtualisointiympäristö.</p> <p>Tässä opinnäytetyössä tutkittiin KVM:n keskeisiä teknologioita ja käytäntöjä. Työssä testattiin KVM:ää komentorivin ja Proxmox VE:n kautta lähestyen. Opinnäytetyöhön on kirjattu näiden kahden erilaisen käyttötavan keskeiset toimenpiteet ja komennot. Työssä käydään läpi muun muassa, miten verkko- ja tallennusratkaisut kannattaa hoitaa KVM-virtualisointia käytettäessä. Työn perusteella pystyy halutessaan luomaan samankaltaisen KVM-virtualisointiympäristön.</p> <p>Teknisesti hypervisorina KVM on kaupallisia kilpailijoita vastaava. KVM ei itsessään sisällä mitään työkaluja, mutta siihen voidaan lisätä tarvittavat ominaisuudet kolmannen osapuolen rajapinnoilla, ohjelmistoilla ja käyttöliittymillä. Periaatteessa KVM-virtuaalikoneita voidaan suorittaa millä tahansa modernilla x86-arkkitehtuurin tietokoneella, jossa on Linux-käyttöjärjestelmä, mutta käytettävyys on pitkälti kolmannen osapuolen työkalujen varassa. KVM-virtualisointialustaa kannattaakin lähestyä jonkin hallintatyökalun kautta. Proxmox VE on yksi tunnetuimmista hallintatyökaluista, joka keskittyy virtualisointialustan hallintaan ja on testien perusteella helppo asentaa ja käyttää. Komentoriviltä KVM:ää ei kannata käyttää muussa kuin testaus- ja harjoituskäytössä tai osana jotain laajempaa ns. pilvialustaa.</p>	
Kieli	Suomi
Asiasanat	KVM, hypervisor, virtualisointi, Proxmox VE
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Business Information Technology
Author(s) Kopeli, Timo	
Title KVM Virtualization via Command Line Interface and Proxmox VE	
Optional Professional Studies Systems Administration	Commissioned by
	Instructor(s) Partanen, Timo
Date Spring 2014	Total Number of Pages and Appendices 50
<p>Kernel-based Virtual Machine or KVM, in short is a relatively new rising technology in virtualization environments that has come to compete with commercial technologies. KVM is open source software and integrated into the Linux kernel, thus it is free to use. Virtualization technology without licensing costs is sure to grab attention of any organization that has licenses taking a big part of their IT budget. This thesis takes a look at the current status of KVM and its technologies and how to build a KVM virtualization environment.</p> <p>KVM's core technologies and practices were studied in this thesis via the command line interface environment and Proxmox VE product. Key practices and commands regarding subjects like networking and storage of these two different approaches can be found in the thesis. With this thesis one can create a KVM virtualization environment for testing and evaluation.</p> <p>As a hypervisor technology KVM matches its commercial competitors. Although KVM itself does not offer many tools or features, necessary ones can be added through third-party software, APIs and management tools. In theory, KVM virtual machines can be run on almost any modern x86 compatible computer running the Linux operating system, but for managing and necessary features one needs a bit more software. It is recommended to approach KVM virtualization via some management tool that suits your needs, such as Proxmox VE which is one of the most popular ones that concentrates on managing of the virtualization infrastructure. It has proven to be easy to use and install. Using KVM via the command line interface, however, cannot be recommended for anything else but testing or experimenting, or as part of bigger cloud-based environment.</p>	
Language of Thesis      Finnish	
Keywords	KVM, hypervisor, virtualization, Proxmox VE
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## SISÄLLYS

1 JOHDANTO	1
2 KVM	3
2.1 KVM:n historia	4
2.2 Tuki vieras-käyttöjärjestelmille	4
2.3 Levyjärjestelmä	4
2.4 Teknologiat	5
2.4.1 QEMU	5
2.4.2 libvirt	5
2.4.3 VirtIO	6
2.5 Työkalut ja käyttöliittymät	7
3 KVM KOMENTORIVILTÄ	8
3.1 Laitteistovaatimukset	8
3.2 Käyttöjärjestelmä	8
3.3 Asennusprosessi	9
3.4 Käyttöliittymä	10
3.4.1 Virsh	10
3.4.2 Virt-manager	10
3.4.3 SPICE ja VNC	11
3.5 Tallennus	11
3.5.1 iSCSI	11
3.5.2 OCFS2	13
3.5.3 LVM	14
3.5.4 NFS	17
3.6 Verkko	17
3.6.1 Bonding	18
3.6.2 Linux bridge	19
3.6.3 Open vSwitch	20
3.7 Virtuaalikoneiden hallinta	22
3.7.1 Virtuaalikoneen luominen	22
3.7.2 Virtuaalikoneen muokkaaminen	23
3.7.3 Virtuaalikoneen poistaminen	24

3.8 Migraatio	24
3.9 Snapshot	25
3.9.1 Virsh	25
3.9.2 LVM	26
3.10 Virtuaalikoneen tuonti toisesta ympäristöstä	27
3.11 High Availability	27
4 PROXMOX VE	28
4.1 Subscription	28
4.2 Teknologiat	29
4.3 Laitteistovaatimukset	29
4.4 Asennusprosessi	30
4.4.1 Klusterin luominen	31
4.4.2 Rekisteröintimuistutuksen poistaminen	31
4.4.3 Repositorion vaihtaminen	31
4.5 Tallennus	32
4.5.1 iSCSI + LVM	32
4.5.2 NFS	34
4.6 Verkko	34
4.6.1 Linux Bonding ja Bridge	35
4.6.2 Open vSwitch bonding ja bridge	36
4.7 Virtuaalikoneiden hallinta	37
4.7.1 Virtuaalikoneen luonti	38
4.7.2 Virtuaalikoneen muokkaaminen	38
4.7.3 Virtuaalikoneen poistaminen	40
4.8 Migraatio	40
4.9 Snapshot ja Backup	42
4.10 High Availability	43
5 POHDINTA	46
LÄHTEET	49

## SYMBOLILUETTELO

best practice	Suosittelu käytäntö
block device	Levylaite, jota käytetään suoraan ilman tiedostojärjestelmää
bonding	Kahdennettu verkkosovitinteknologia
bridge	Silta-tilan verkkosovitin
distribuutio	Käyttöjärjestelmän jakelu
HA	High Availability, korkea järjestelmän saatavuus
hosti	Fyysinen virtualisointipalvelin
hypervisor	Virtualisointityökalu
klusteri	Usean tietokoneen samaa toimintoa suorittava ryhmä
KVM	Kernel-based Virtual Machine, hypervisor
LVM	Logical Volume Manager, looginen tilinhallinta
migraatio	Virtuaalikoneen siirtäminen toiselle fyysiselle palvelimelle
node	Klusteriin liitetty fyysinen tai virtuaalinen kone
repositorio	Ohjelmistopakettien tietokanta
snapshot	Tilannevedos virtuaalikoneesta
storage	Tallennusratkaisu
volume group	LVM:n tiliryhmä

## 1 JOHDANTO

Virtualisointi on jo vuosia ollut palvelinympäristöissä laajassa käytössä oleva tekniikka, jolla voidaan pyörittää useita palvelimia pienellä määrällä fyysisiä laitteita. Jos olet kuullut puhut-tavan pilvistä tietotekniikan yhteydessä, niin virtualisointitekniikka on pilvien takana. Markkinoilla on suuria kaupallisia virtualisointialustojen kehittäjiä, kuten VMware ja Microsoft, joiden alustat ovatkin kehittyneet paljon kilpailun ansiosta.

Viime vuosien aikana markkinoilla on nostanut päätään avoimen lähdekoodin projekti Kernel-based Virtual Machine, lyhyesti KVM. Se on kasvattanut suosiotaan nopeasti, varsinkin suurten pilvipalveluita ja virtuaalipalvelimia tarjoavien yritysten keskuudessa. GPL-lisensoidun KVM:n ilmaisuus ja avoin lähdekoodi takaavat jatkuvan ja nopean kehityksen, ja onkin helppo nähdä, miksi se olisi suuri kiinnostuksen kohde kaikille, joilla on kuluja virtualisointialustan lisenssimaksuista.

Tämän opinnäytteen tavoitteena on tutkia KVM:n tämänhetkistä tilannetta ja tekniikoita sekä sitä, kuinka sen avulla voidaan pystyttää palvelinvirtualisointiympäristö. Tavoitteena on myös, että tämän opinnäytteen avulla KVM virtualisoinnista kiinnostunut voi tehdä valintoja tekniikoiden välillä ja jopa pystyttää oman virtualisointiympäristönsä.

Tarkastelen työssä KVM:ää enemmän teknisestä näkökulmasta, enkä lähtenyt analysoimaan käyttöliittymiä tai vertaamaan niitä kilpailijoiden tuotteisiin, sillä samaan aikaan kanssani samalla laitteistolla juuri näitä asioita tutki Tarmo Kääriäinen omaan opinnäytteeseensä KVM-hallintatyökalut. Hän käy työssään läpi KVM:n hallintatyökaluista Proxmox VE:n lisäksi OpenNode-, Eucalyptus-, oVirt- ja OpenNebula hallintatyökalut ja vertailee niitä keskenään.

KVM:n rakenne on vähän erilainen, kuin kaupallisissa virtualisointialustoissa. Se ei itsessään tarjoa hallintatyökaluja, vaan siinä tukeudutaan täysin kolmannen osapuolen ratkaisuihin. Hallintatyökaluja on olemassa useampi kourallinen ja osa niistä on osittain maksullisia, mutta tässä työssä käytettiin vain ilmaisessa jakelussa olevia tuotteita ja ohjelmistoja. Hallintatyökalujen vertailu ei ole tämän opinnäytetyön keskeinen aihe, vaan KVM-virtualisointia lähestytään komentorivin ja suositun Proxmox VE -tuotteen kautta.

Opinnäytetyö perustuu pitkälti tekemiini KVM-testauksiin. Komentoriviltä käytettävää KVM:ää ajettiin Ubuntu Server 13.10 distribuution päällä. Kokeilin myös Fedora ja CentOS

distribuutioita, mutta pääasialliseksi alustaksi päätin valita Ubuntun, koska se on minulle tuttu järjestelmä. Proxmox VE:tä testattiin versioilla 3.1 ja 3.2.

Proxmox VE:llä ajettiin testejä kahdessa vaiheessa. Fyysisellä laitteistolla testattiin 3.1-version High Availability ja klusterointi ominaisuuksia syksyllä 2013. Keväällä 2014 suoritettiin 3.2-versiolla virtuaaliympäristössä lisätestauksia, missä testattiin muun muassa Open vSwitch -tukea. Komentorivin KVM-testejä suoritettiin syksyn 2013 ja kevään 2014 aikana.

Fyysisenä laitteistona käytettiin pääasiassa Kajaanin ammattikorkeakoulun käyttöjärjestelmälaboratorion Dell PowerEdge R410-sarjan palvelimia, joissa on kaksi Intel Xeon 5500-sarjan prosessoria ja 16 Gigatavua RAM-muistia. Osassa testeistä käytettiin myös VMware ESXi 5.1:n päällä ajettuja virtuaalikoneita. Tallennusjärjestelmänä käytössä oli EMC CX3 iSCSI-levyjärjestelmä.

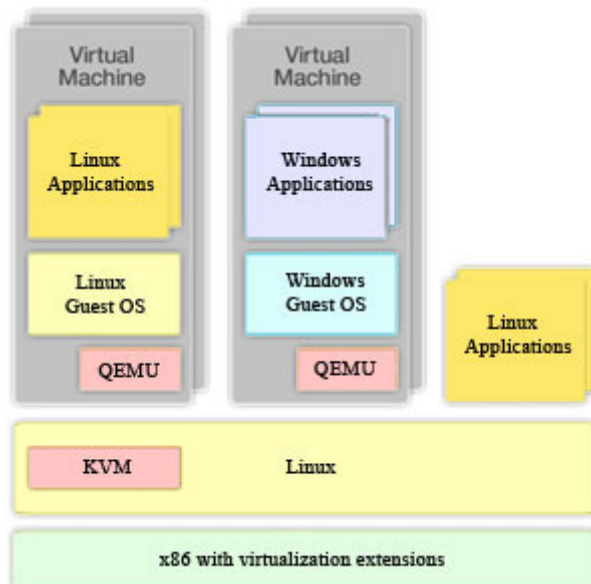
Lähdeaineistona on käytetty pääasiassa erilaisia Internet-lähteitä. Komennoissa ja konfiguraatioesimerkeissä ympäristökohtaiset muuttujat on kirjoitettu lihavoidulla fontilla.



## 2 KVM

KVM on lyhenne englanninkielisistä sanoista Kernel-based Virtual Machine, joka tarkoittaa Linux-ytimeen rakennettua tukea virtualisoinnille. KVM:n avulla Linux pystyy toimimaan hypervisorina ja suorittamaan virtuaalikoneiden käskyjä suoraan prosessorilla. (Wikipedia 2014)

KVM toimii x86-arkitehtuuria tukevilla prosessoreilla ja vaatii prosessorilta Intel VT-x tai AMD-V -tuen. (KVM a 2014)



Kaavio 1. KVM:n rakenne. (IBM a 2014)

KVM suorittaa virtuaalikoneita prosesseina Linux-käyttöjärjestelmässä ja toisen avoimen lähdekoodin sovelluksen QEMU:n avulla virtuaalikoneiden prosesseille voidaan emuloida erilaisia I/O-laitteita. Virtuaalikoneet voivat suorittaa Linux-, Windows- ja BSD-pohjaisia käyttöjärjestelmiä sekä muutamia muita tuntemattomampia.

## 2.1 KVM:n historia

KVM:n on alun perin koodannut israelilainen ohjelmistokehittäjä Avi Kivity työskennellessään Qumranet-yhtiölle vuonna 2006. Se integroitiin Linux-kerneliin vuoden 2007 alussa Linux-kernel versiossa 2.6.20. (Wikipedia 2014)

eWeek:n tekemässä haastattelussa Kivity kertoo, että KVM:n saaminen Linux-kerneliin oli helppoa, koska aika oli juuri oikea projektille. Virtualisointi oli kuuma juttu siihen aikaan ja ainoa vapaan lähdekoodin vaihtoehto oli Xen, joka toimii Linux-kernelin ulkopuolella. (eWeek 2013)

Red Hat osti Qumranetin vuonna 2008 hintaan 107 miljoonaa dollaria. Red Hat onkin ottanut KVM:n oman kaupallisen RHEV-virtualisointialustansa perustaksi. KVM:ää ylläpitävät nykyään Paolo Bonzini ja Gleb Natapov. (Wikipedia 2014)

## 2.2 Tuki vieras-käyttöjärjestelmille

KVM:n käyttöjärjestelmätuki vieraskoneille ei ole niin hyvin kartoitettu kuin esim. VMwaren. Käytännössä KVM tukee kaikkia nykyaikaisia käyttöjärjestelmiä, kuten eri versioita Windows, Linux, Solaris ja BSD -käyttöjärjestelmistä. (KVM b 2014)

KVM:n vieras-käyttöjärjestelmien tuesta on koottu taulukkoa osoitteessa [http://www.linux-kvm.org/page/Guest\\_Support\\_Status](http://www.linux-kvm.org/page/Guest_Support_Status) .

## 2.3 Levyjärjestelmä

KVM:n kanssa voidaan käyttää useita erilaisia tallennusratkaisuja. IBM:n best practicen mukaan paras suorituskyky saavutetaan käyttämällä erillisistä block deviceä jokaiselle virtuaalikoneelle, ilman välissä toimivia tiedostojärjestelmiä. (IBM b 2014)

Levyjärjestelmän valinta onkin hyvin pitkälti ylläpitäjän maku- ja budjettiasia. Yleisin ratkaisu lienee LVM-osiot käyttäen ulkoista levyjärjestelmää iSCSI:n kautta. Toiset taas suosivat NFS:ää, jos haluavat käyttää virtuaalikoneiden kiintolevyjä tiedostomuodossa.

Käytännön testeissä todettiin, että käytettäessä LVM block devicejä virtuaalikoneen kiintolevyinä kirjoitus- ja lukunopeus ovat nopeampia kuin käytettäessä tiedostomuodossa olevia kiintolevyjä. Myös haku aika levyille oli noin neljä kertaa nopeampi.

## 2.4 Teknologiat

KVM hyödyntää useita kolmannen osapuolen teknologioita eikä itsessään ole kokonaisvaltainen ratkaisu. Perustoimenpiteidenkin suorittaminen olisi hyvin vaikeaa ilman näitä muita teknologioita.

### 2.4.1 QEMU

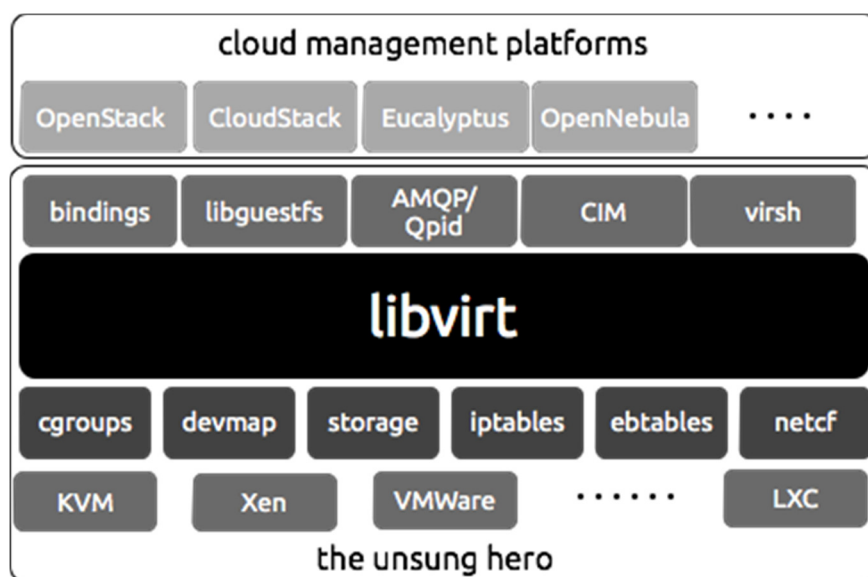
Vaikka KVM:ää yleensä kutsutaan virtualisointialustaksi ja sitä se onkin, se ei yksin kykene täyteen konevirtualisointiin. KVM eristää virtuaalikoneelle alueen fyysiseltä koneelta, mutta se ei itsessään emuloi erilaista rautaa virtuaalikoneelle, vaan tarjoaa samanlaisen fyysisen koneen kaltaisen alustan, jossa muut ohjelmat voivat suorittaa toimintoja. QEMU hyödyntää tätä KVM:n tarjoamaa alustaa suorittamalla virtuaalikoneiden prosessit KVM:n eristämässä lohkoissa.

QEMU:n avulla voidaan emuloida useita erilaisia prosessoreita ja jopa eri arkkitehtuureja. KVM onkin nykyään yleensä paketoitu muokatun QEMU:n kanssa samaan pakettiin, jolla on yhteinen suoritettava tiedosto: qemu-kvm. Se on saatavilla tunnetuimpien Linux distributioiden repositorioista.

### 2.4.2 libvirt

Libvirt on rajapinta ja kokoelma sovelluksia, palveluita ja kirjastoja, jotka tuovat tarpeellisia ominaisuuksia KVM:n ja QEMU:n virtualisointiin. Esimerkiksi verkotus ja virtuaalikoneiden laitteiston hallinta tapahtuu libvirtin kautta. Ilman libvirtä voidaan kyllä käyttää KVM:ää, mutta silloin menetetään suuri osa hallintatyökaluista ja ominaisuuksista. Jotkin kaiken katta-

vat valmiit järjestelmät kyllä käyttävät KVM:ää suoraan, mutta niillä on omat sovellukset hoitamaan asioita, jotka libvirt normaalisti hoitaa.



Kaavio 2. Libvirtin sijoittuminen virtualisoinnin rajapintana. (Vyom 2013)

Asetuksensa libvirt tallentaa selkeässä xml-formaatissa. Asetustiedostoja voidaan muokata käsin tai esimerkiksi virsh-ohjelmalla. Libvirtin xml-formaatin muuttujat on selitetty perusteellisesti sen kotisivuilla osoitteessa <http://libvirt.org/format.html>.

Libvirt tukee KVM:n lisäksi myös muita virtualisointialustoja, kuten Xen, LXC ja VMware ESXi.

### 2.4.3 VirtIO

VirtIO on standardoitu virtuaalikoneiden laitteistoajuri, joka tarjoaa muuten täysin virtualisoiduille virtuaalikoneille paravirtualisoituja verkko- ja tallennuslaitteita. KVM voi emuloida esimerkiksi oikeaa RTL8139-verkkokorttia, mutta se tuottaa turhaa kuormaa hypervisorille. VirtIO:ta käyttämällä virtuaalikoneelle kerrotaan rehellisesti, että laite on virtualisoitu, jolloin saavutetaan parempi suorituskyky.

Linux-kernel versiot 2.6.25 ja uudemmat tukevat VirtIO:ta, joten se on usein käytetty Linux-virtuaalipalvelinten kanssa. Windows-käyttöjärjestelmät eivät tue VirtIO:ta ilman erikseen asennettavia ajureita.

## 2.5 Työkalut ja käyttöliittymät

Viime vuosien aikana on markkinoille tullut useita erilaisia hallintatyökaluja ja käyttöliittymiä, jotka on tarkoitettu KVM:n hallintaan tai jotka tukevat KVM:ää muiden hypervisoreiden lisäksi.

KVM:n hallintatyökalut voidaan jaotella teknologian perusteella kahteen ryhmään: libvirtia rajapintana käyttävät ja jotain muuta rajapintaa käyttävät. Suurin osa työkaluista luottaa libvirtiin, mutta jotkin kehittäjät ovat luoneet omat rajapintansa ja ohjelmansa hoitamaan samoja toimintoja.

Työkalut voidaan myös jakaa kahteen ryhmään myös ideologian mukaan. Osa työkaluista on perinteisen virtuaaliympäristön hallintaan tarkoitettuja ja osa on suunniteltu pilviympäristöjä pyörittämään. (Kääriäinen 2014)

Suuri osa työkaluista, varsinkin libvirtia käyttävistä, asennetaan KVM:ää tukevaan Linux-käyttöjärjestelmään erillisenä pakettina. On myös olemassa valmiiksi paketoituja ns. bare-metal-virtualisointi distribuutioita, joiden asennuskuva sisältää Linux-ytimen ja etukäteen suurimmilta osin konfiguroidun KVM-virtualisointialustan yhteen paketoituna jonkin hallintajärjestelmän kanssa.

Käyttöliittymät ovat suurimmilta osin web- ja komentorivipohjaisia. Erillisiä työasemalla ajettavia hallintaohjelmia ei juurikaan ole. Graafisen konsolilyhteyden luomiseksi käytetään useimmiten VNC- tai SPICE-teknologioita, jotka toki vaativat työasemalta asianmukaiset ohjelmistot.

KVM:n käyttöliittymiä tulee jatkuvasti lisää ja niistä voi löytää listauksen KVM:n epävirallisilta Internet-sivuilta osoitteesta [http://www.linux-kvm.org/page/Management\\_Tools](http://www.linux-kvm.org/page/Management_Tools).

### 3 KVM KOMENTORIVILTÄ

Ns. raakaversio-KVM:ssä itsessään on rajoitetusti ominaisuuksia, mutta erinäisten pakettien avulla niitä on hyvinkin paljon. Osa vaatii kuitenkin monimutkaisia toimenpiteitä toimiakseen. Tässä luvussa tarkastellaan, miten ne on toteutettu.

#### 3.1 Laitteistovaatimukset

KVM ei ole nirso laitteiston suhteen. Sen vaatimuksena on vain prosessorin tuki virtualisoinnille, AMD-V- tai Intel VT-muodossa. Toisena vaatimuksena voitaisiin nimetä Linux kernelin tuki KVM:n moduulille. Koska kaikki nykyaikaiset Linux-distribuutiot käyttävät Linux kernel 2.6.20 -versiota, jossa KVM-moduulit sisäistettiin kerneliin, tai selvästi uudempaa versiota, ei asiaa voi pitää enää relevanttina. Muistin määrä ja prosessorin suoritusteho vaikuttavat vain suorituskyyneen eivätkä rajoita käyttöönottoa kokonaisuudessaan. KVM toimii sekä 32- että 64-bittisellä tietokoneella, mutta 32-bittisellä koneella ei voi virtualisoida 64-bittistä virtuaalikoneetta. (KVM a 2014)

KVM:n laitteistovaatimukset ovatkin siis paljon vähäisemmät kuin esimerkiksi VMwaren ESXi 5.1:ssä, jossa vaatimuksena on prosessorin virtualisointituen lisäksi 64-bittinen prosessori, 2 prosessoriydintä, 2 gigatavua RAM-muistia ja tuettu vähintään 1 gigabitin verkkokortti. (VMware 2014)

#### 3.2 Käyttöjärjestelmä

KVM:n käyttöön tarvitaan jokin Linux-distribuutio. Valikoima on laaja ja suuressa osassa virtualisointi on mahdollista. Tunnetuimpia lienevät Ubuntu, Debian, Fedora, CentOS tai RedHat Enterprise Linux. Tunnetuimpien distribuutioiden kanssa pärjää paremmin ongelmatilanteiden sattuessa, mutta yleisesti ottaen kannattanee käyttää sitä distribuutiota, josta on jo etukäteen kokemusta. Komennot ja toimintatavat vaihtelevat eri distribuutioiden välillä. Varsinkin Debian-pohjaiset distribuutiot ovat hyvin erilaisia RedHat-pohjaisiin verrattuna.

Koska olen käyttänyt pääasiassa Ubuntuä, käytin sitä myös testeissäni. Tärkeimmät komponentit ovat paketit qemu-kvm, libvirt-bin, virsh ja lvm2. Muissa distribuutioissa, varsinkin RedHat-pohjaisissa pakettien nimet voivat olla hieman erilaiset.

### 3.3 Asennusprosessi

KVM:n ja muiden tarvittavien asennus on hyvin helppo. Se löytyy nykyaikaisten Linux-distribuutioiden repositorioista. Esimerkiksi Ubuntuissa on myös asennusvaiheessa vaihtoehto, jossa virtualisointialustan voi asentaa automaattisesti. Se asentaa qemu-kvm ja libvirt -paketit sekä niiden mahdollisesti tarvitsemat riippuvuudet. Fedoraissa ja CentOSissa on myös samankaltainen ominaisuus.



Kuva 1. Ubuntu Linuxin asennuksen yhteydessä näkyvä toimintovalintaikkuna.

Asennuksen jälkeen alusta on käyttövalmis ja siihen voi yhdistää virsh:llä tai virt-managerilla. IP-osoitteen voi vaihtaa distribuution normaaleilla työkaluilla. Muiden käyttöliittymien käyttäminen edellyttää yleensä niiden asentamista erikseen.

Valmiiksi pakatut virtualisointialustadistribuutiot, kuten Proxmox VE, ovat paljon yksinkertaisempia ja suoraviivaisempia asentaa. Asennusohjelma kysyy kutakuinkin vain verkkoasetukset ja pääkäyttäjän salasanan, minkä jälkeen järjestelmä on valmis käytettäväksi.

### 3.4 Käyttöliittymä

Vakiona KVM:n mukana ei varsinaisesti ole mitään käyttöliittymää, vaan qemu-kvm-ohjelmalle annetaan parametreja komentoriviltä. Käytännössä aina jos käytetään KVM:ää komentoriviltä, mukaan on niputettu libvirt-rajapinta ja Virsh-komentoriviohjelma, jolla voidaan yksinkertaisilla komennoilla hallita ja muokata virtuaalikoneiden asetuksia. Virt-manager on graafinen versio Virshistä.

#### 3.4.1 Virsh

Vaikka KVM:llä, QEMU:lla tai libvirt:llä ei periaatteessa ole “omaa” hallintasovellusta, Ubuntun, CentOS:n ja monen muunkin distribuution automaattinen virtuaalialustan asennus asentaa siihen tarkoitukseen virsh-ohjelman. Virshiä voidaan käyttää hallinnoimaan sekä paikallisia että etäkohteessa sijaitsevia virtuaalikoneita.

Virsh on komentorivisovellus. Sillä voidaan hallita kaikkia libvirtin tukemia virtualisointialustoja. Sen komennot ovat yksinkertaisia ja sen avulla voidaan suorittaa monia toimintoja, kuten virtuaalikoneiden käynnistys ja sammutus, migraatio, virtuaaliverkkojen luonti ja virtuaalikoneen laitteiston hallinta. Virtuaalikoneiden laitteiston hallinta tapahtuu XML-tiedostoja muokkaamalla.

#### 3.4.2 Virt-manager

Virt-manager, tai VMM eli Virtual Machine Manager, on käytännössä graafinen versio virsh:stä. Sillä voi kätevästi käynnistää ja sammuttaa sekä muokata virtuaalikoneiden asetuksia ilman tietämystä libvirtin xml-syntaksista. Koska molemmilla on sama kehittäjätiimi, kukaunkin kaikki toiminnot, esimerkiksi migraatiot, jotka voidaan tehdä virshillä, voidaan tehdä myös virt-managerilla. Virt-managerissa on myös integroitu virt-viewer, VNC ja SPICE graafinen katselusovellus, joka helpottaa virtuaalikoneiden asennusta huomattavasti.



### 3.4.3 SPICE ja VNC

Virtuaalikoneiden käyttöjärjestelmiä asennettaessa tarvitsee yleensä jollain tavalla kirjoittaa ja katsella asetuksia, mutta virtuaalikoneella ei ole omaa näyttöä tai näppäimistöä. KVM tukee kuitenkin kahta tapaa välittää kuvaa ja komentoja käyttäjän ja virtuaalikoneen välillä. VNC on kahdesta se ns. tuttu ja turvallinen, se ei vaadi mitään erinäisiä ohjelmia hostille ja sen libvirtin kautta tehdyt koneet saavatkin vakiona. Toinen vaihtoehto SPICE, joka on uudempi ja nopeampi, mutta sen käyttö vaatii hostille asennettavan spice-paketin, virtuaalikoneelta QXL -grafiikka-ajurit ja client-koneelta SPICE-katselusovelluksen.

Virt-manager ja virt-viewer tukevat sekä VNC:tä että SPICE:ä libvirtin kautta. Virtuaalikoneisiin voidaan yhdistää myös suoraan VNC-asiakasohjelmilla tai SPICE:n remote-viewerillä, jos tiedetään missä hostin portissa ne löytyvät.

## 3.5 Tallennus

Levyjärjestelmänä voidaan käyttää lähes kaikkia mahdollisia tallennusratkaisuja, kunhan ne saadaan näkymään Linuxissa kiintolevyinä tai muuten liitettyä niin, että niihin voi tallentaa.

Yleisimmät tavat ovat käyttää NFS-verkkotallennusta tiedostomuotoisille virtuaalikoneille tai LVM block device muodossa iSCSI:n yli.

### 3.5.1 iSCSI

iSCSI on teknologia jolla lähetetään SCSI käskyjä ja dataa Ethernet-verkon yli. Sillä voidaan liittää erillisessä verkkotallennusratkaisussa sijaitsevia levyjakoja, LUN:eja, host-koneisiin. Näiden liitettyjen levyjen tallennustilaa voidaan sitten hyödyntää kuin ne olisivat fyysisesti koneeseen liitettyjä kiintolevyjä. Linuxissa iSCSI:tä hoitaa open-iscsi-palvelu.

iSCSI:tä voidaan myös hyödyntää käyttämällä LUN:eja suoraan per virtuaalikone. Ratkaisun pätevyys riippuu paljon levyjärjestelmän hallintaominaisuuksista ja käytettävyydestä. Jos käytössä on jokin vanha levyjärjestelmä, jossa käyttöliittymä on buginen ja kankea, ei tee mieli

käyttää sitä päivittäiseen työskentelyyn. Uudemmissa järjestelmissä, kuten IBM SVC:ssä tilanne on toki toisin ja ratkaisu on voi olla varsin pätevä.

Yksi tapa hyödyntää iSCSI:n tarjoamia levyjä on suurten LUN:ien teko, joihin tehdyt tiedostojärjestelmät liitetään host koneeseen johonkin hakemistoon, joita sitten käytetään libvirtin pooleina. Tällä tavalla virtuaalikoneiden kiintolevyt ovat tiedostoja, eivätkä block devicejä. Jos käytössä on vain yksi host-kone, tiedostojärjestelmäksi käy mikä tahansa Linuxin tukema, mutta jos kyseessä on klusteri, tarvitaan jokin tiedostojärjestelmä joka tukee klusterointia. Ilman klusteroitutukea tiedostojärjestelmässä tallennettu tieto voi korruptoitua usean hostin yrittäessä kirjoittaa yhtäaikaaisesti.

iSCSI:n käyttö vaatii hostilta iSCSI Initiatorin. Sen paketit löytyvät repositoriosta.

```
apt-get install open-iscsi open-iscsi-utils
```

Initiatorilla pitää olla konekohtainen nimi, joka muokataan `/etc/iscsi/initiatorname.iscsi` -tiedostoon. Nimen tulee olla IQN-standardin mukainen. Alkuosa on hyvä jättää sellaiseksi kuin se on valmiina ja muokata vain ensimmäisen kaksoispisteen jälkeen tulevaa osaa.

```
InitiatorName=iqn.1993-08.org.debian:05:KVM5INIT
```

Jotta iSCSI liitettävät nodet lähtevät päälle automaattisesti hostin käynnistyessä täytyy muokata `/etc/iscsi/iscsid.conf` -tiedostoa.

```
...
node.startup = automatic
...
```

iSCSI targettien osoitteet ovat yleensä hyvin pitkiä IQN-nimiä, mutta ne löytyvät helposti verkosta portaalin avulla, joka on yleensä levyjärjestelmän IP-osoite ja portti 3260.

```
sudo iscsiadm -m discovery -t st -p 172.31.17.24:3260
```

```
sudo iscsiadm -m node --login
```

iSCSI targetit voidaan targetit lisätä myös manuaalisesti.

```
iscsiadm -m node -T iqn.1992-04.com.emc:cx.ck200081800112.a0 -p
172.31.17.23:3260 -o new
```

```
sudo iscsiadm -m node --login
```

Linux-hostin laitteista pitäisi löytyä login-komennon jälkeen fyysisiä levyjä, joita voidaan käyttää virtuaalikoneiden levyinä tai vaikka LVM:n alustana.

### 3.5.2 OCFS2

Yksi klusteroitu tiedostojärjestelmä vaihtoehto on Oraclen OCFS2, joka on ilmainen ja tuettu Linux-kernelissä vakiona. Sen tarvitsemat hallintatyökalut ja paketit löytyvät useimpien distribuutioiden repoista. OCFS2 mahdollistaa saman iSCSI levyn käytön useassa host-koneessa yhtäaikaaisesti tiedostojärjestelmämuodossa. Jokaiselle host-koneelle pystytetään OCFS2-palvelu, joka pitää yhteyttä muihin hosteihin ja koordinoi kirjoitusoikeuksia.

```
sudo aptitude install ocfs2-tools
```

```
sudo dpkg-reconfigure ocfs2-tools
```

OCFS2 käyttää asetustiedostonaan ”/etc/ocfs2/cluster.conf”-tiedostoa, ja sen tulee olla identtinen jokaisella klusteriin kuuluvalla hostilla. Asetustiedoston luonnin jälkeen hostit täytyy käynnistää uudelleen.

```
cluster:
name = ocfs2
node_count = 2
node:
name = KVM1
cluster = ocfs2
number = 0
ip_address = 192.168.200.11
ip_port = 7777
node:
name = KVM2
cluster = ocfs2
number = 1
ip_address = 192.168.200.12
ip_port = 7777
```

Klusteriin voi lisätä nodeja dynaamisesti, mutta poistaminen vaatii koko klusterin alasajon.

Tarvitaan fyysisiä levyjä joille on tehty osio. Osioita voi luoda vaikka fdisk-ohjelmalla. Osiot voivat sijaita jollain jaetulla levyjärjestelmällä, esimerkiksi iSCSI:n yli.

OCFS2 on oma tiedostojärjestelmänsä. Levyosiot pitää alustaa sen tukemaan muotoon.

```
mkfs.ocfs2 /dev/sdc1
```

Automaattista liittämistä uudelleenkäynnistyksessä varten täytyy muokata ”/etc/fstab”-tiedostoa. Voidaan käyttää by-path tai by-id polkuja. Kansiot johon osiot liitetään pitää olla olemassa olevia tyhjiä kansioita.

```
...
/dev/disk/by-path/ip-172.31.17.24:3260-iscsi-iqn.1992-
04.com.emc:cx.ck200081800112.b0-lun-0-part1 /mnt/500POOL ocfs2
_netdev 0 0
...
```

Uudelleenkäynnistyksen jälkeen liitetuille osioille voidaan kirjoittaa yhdeltä hostilta kerrallaan. Osiot voidaan myös liittää ilman uudelleenkäynnistystä ”mount -a”-komennolla.

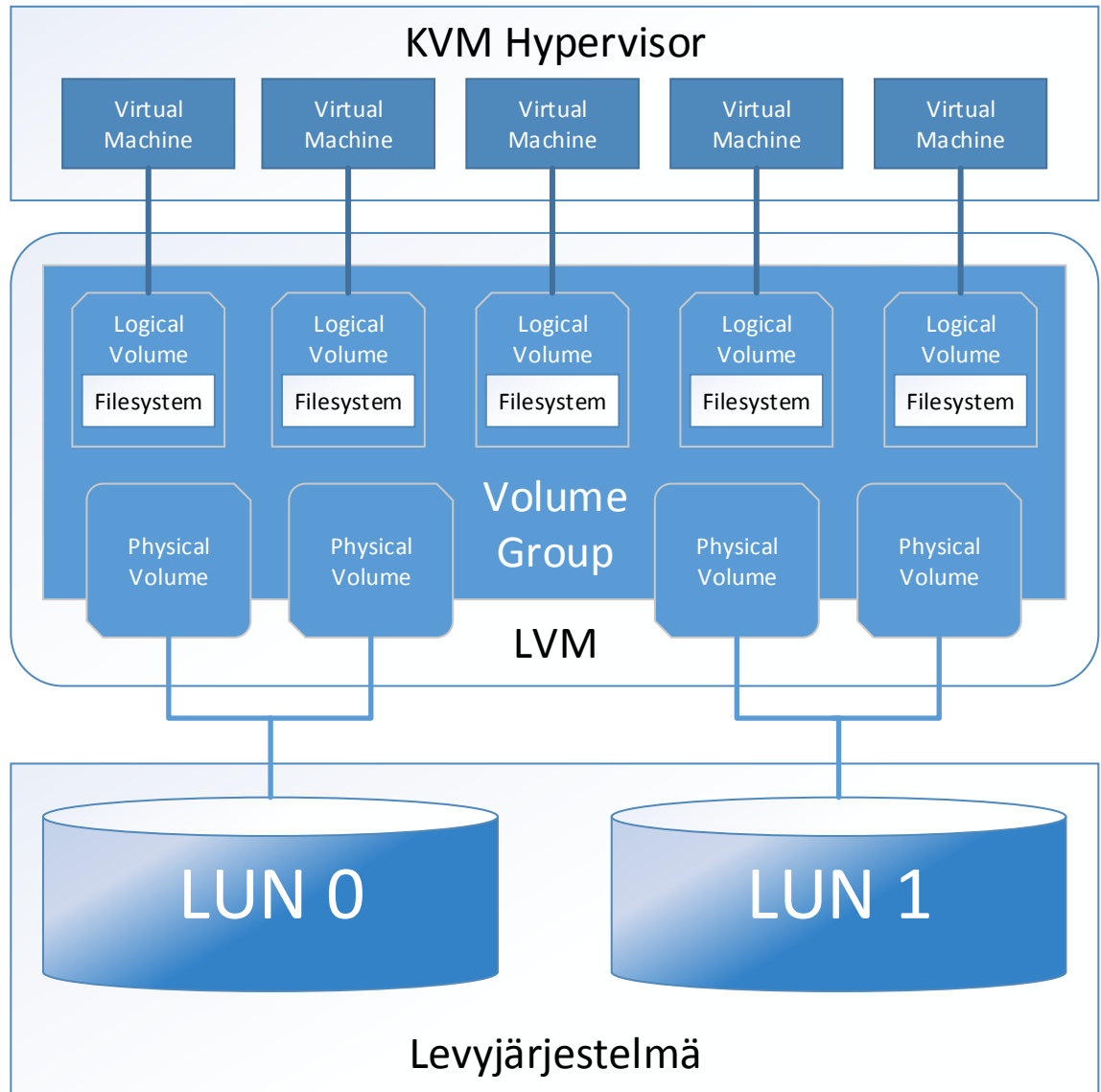
### 3.5.3 LVM

Jos halutaan käyttää block devicejä, mutta levyjärjestelmän käyttöliittymä on epämiellyttävä ja epäkäytännöllinen, voidaan iSCSI:a hyödyntää käyttämällä LVM:ää välissä luomaan loogisia levyjä. Jos käytössä on useampi kuin yksi hosti, tarvitaan cLVM lisäosa, mikä mahdollistaa klusteroinnin LVM:ssä.

LVM, eli Logical Volume Manager on yleisin tapa käyttää virtuaalikoneiden kiintolevyjä block device -muodossa. Se löytyy vakiona käytännössä kaikista Linux-distribuutioista, joten se on helppo valinta. Sillä voidaan näyttää eristettyjä paloja usean fyysisen levyn poolista virtuaalikoneille käytettäväksi. Näitä paloja voidaan luoda, poistaa ja muokata yksinkertaisilla komennoina dynaamisesti ns. lennosta.

LVM:n ideana on, että joukosta Linuxissa näkyviä fyysisiä levyjä luodaan yksi suuri volume group. Volume grouppiin luodaan loogisia levyjä joita sitten allokoidaan virtuaalikoneille.

LVM:ään kelpaavat kaikki Linuxin tukemat kiintolevyt. Näin ollen siihen voidaan liittää esim. iSCSI tai muiden SAN-järjestelmien kiintolevyjä.



Kaavio 3. LVM:n rakenne.

LVM tarvitsee muutaman paketin, jotka löytyvät repositoriosta.

```
apt-get install lvm2 cman clvm
```

Klusterointituki täytyy aktivoida LVM:n asetuksiin.

```
lvmconf --enable-cluster
```

Klusterointia varten täytyy luoda `/etc/cluster/cluster.conf` -tiedosto kaikille klusteriin liitetäville hosteille.

```
<?xml version="1.0"?>
<cluster name="kvmcluster" config_version="4">
  <cman two_node="1" expected_votes="1"/>
</cluster>
```

```

<clusternodes>
  <clusternode name="KVM2.kvm.local" nodeid="1" votes="1">
    <fence>
    </fence>
  </clusternode>
  <clusternode name="KVM5.kvm.local" nodeid="2" votes="1">
    <fence>
    </fence>
  </clusternode>
</clusternodes>
<fencedevices>
</fencedevices>
<rm>
</rm>
</cluster>

```

Aina kun `"/etc/cluster/cluster.conf"`-tiedostoa muokataan manuaalisesti, on muistettava kasvattaa versionumeroa yhdellä kohdassa `"config_version"`. Muokkausten jälkeen ajetaan päivityskomento, jonka jälkeen on hyvä käynnistää hostit uudelleen.

```
cman_tool version -r -S
```

CMAN ja CLVM ovat klusteroidun LVM:n palveluita, jotka hoitavat levyjen organisoinnin ja pitävät huolta, että vain yksi hosti voi kirjoittaa levyille kerrallaan.

LVM tarvitsee fyysisen levyn osioita, joista se rakentaa loogisia pooleja. Näihin pooleihin se luo loogisia levyjä, joita virtuaalikoneet voivat käyttää levyinään. Fyysiset osiot voivat sijaita paikallisesti hostilla tai vaikka iSCSI:n kautta verkossa. Klusteroitua LVM:ää tehtäessä jokin jaettu levyjärjestelmä on pakollinen.

Levyille täytyy tehdä osio ja osioiden tyyppiä täytyy valita Linux LVM. Toiminnon voi suorittaa esimerkiksi `fdisk`-ohjelmalla.

Linux LVM tyyppin osioita voidaan merkitä LVM:n käyttöön `pvcreate`-komennolla. Toimenpide voidaan tehdä kerralla usealle levyosiolle.

```
pvcreate /dev/sdc1 /dev/sdd1
```

Näille osiolle voidaan luoda volume group, johon liitetään yksi tai useampia osioita.

```
vgcreate cx3-vg0 /dev/sdc1 /dev/sdd1
```

Libvirttiin voidaan luoda pooli, joka käyttää luotua volume groupia.

```
virsh pool-create-as cx3-vg0 logical --target /dev/cx3-vg0
```

```
virsh pool-autostart cx3-vg0
```

Virsh, virt-manager ja virt-install voi luoda libvirtin pooliin uusia levyjä ja poistaa niitä.

```
virsh vol-create-as cx3-vg0 test1 16G
```

Loogisia LVM-osioita voidaan myös käyttää virtuaalikoneissa manuaalisesti, ilman libvirttiä.

### 3.5.4 NFS

Jos iSCSI ei sovi tai miellytä, yksinkertaisempi ratkaisu voi olla parempi. Virtuaalikoneiden levyt tehdään image-tiedostoiksi, joita liitetään virtuaalikoneille. Nämä imagetiedostot voivat sijaita käytännössä missä tahansa tallennuspaikassa, joka on luettavissa ja kirjoitettavissa host-koneelta. Yksi yleisimmistä tavoista on käyttää NFS-verkkotallennusta, koska useat levyjärjestelmät tukevat sitä vakiona ja se on helppokäyttöinen. Oman NFS-palvelimen pystytys on myös erittäin yksinkertaista, jos käytössä ei ole erillistä levyjärjestelmää.

NFS on helppokäyttöinen. Libvirttiin luodaan NFS-pooli, joka käytännössä liittää sen vain johonkin hostin kansioon automaattisesti.

```
virsh pool-create-as nfs-pool netfs --source-host 192.168.200.18  
--source-path /kvmjako --target /mnt/nfs-pool
```

```
virsh pool-autostart nfs-pool
```

Virsh, virt-manager ja virt-install voi luoda libvirtin pooliin uusia levyjä ja poistaa niitä.

```
virsh vol-create-as cx3-vg0 test1 16G
```

NFS-poolissa virtuaalikoneiden levyt ovat tiedostomuotoiset, joten niitä voidaan kopioida ja siirtää helposti tarvittaessa.

### 3.6 Verkko

KVM:n verkko-ominaisuudet ovat vakiolta välttävät. Libvirtin vakio NAT-verkko tarjoaa hostin sisäisen verkon, jolla pääsee internettiin, mutta ulko verkkoon tarjottavien palveluiden pyörittäminen NAT-verkossa ei ole käytännöllistä. Linux-kernel tukee kuitenkin siltausta ja

ei vaadi kuin hostin verkkoasetusten muokkaamista. Siltauksella saadaan virtuaalikone jakamaan hostin fyysinen verkkokortti sulassa sovussa ja virtuaalikoneiden verkotus voidaan hoitaa kuin ne olisivat fyysisiä koneita kaapelilla kiinni kytkimessä.

Jos halutaan edistyneempiä ominaisuuksia, kuten usean hostin yhteisiä sisäisiä verkkoja, tarvitaan jokin erillinen ratkaisu. Esimerkiksi Open vSwitch on yksi tällainen edistyksellisempi sovellusratkaisu. Libvirt tukee sitä, mutta valitettavasti kirjoitushetkellä mitkään käyttöliittymät eivät, joten sen käyttö vaatii virtuaalikoneiden asetustiedostojen manuaalista muokkaamista. Tämä hankaloittaa uusien koneiden luomista, mutta on kuitenkin toimiva ratkaisu, jos manuaalinen muokkaaminen ei ole ongelma.

Virtuaalikoneille voidaan myös määrittää fyysinen verkkokortti hostilta. Näin tehtäessä suorituskyky vastaa fyysisen kortin ominaisuuksia. Tätä verkotustapaa käytettäessä tarvitaan useita verkkokortteja per fyysinen hosti, joten sitä ei yleensä käytetä kuin todella tärkeille virtuaalikoneille.

Yleisin tapa verkottaa lienee käyttää siltausta ja fyysisiä verkkokortteja tarpeen mukaan ja hoitaa tarkempi verkotus kytkinten puolella käyttäen hyväksi esimerkiksi VLAN:ejä ja muita kytkinten ominaisuuksia verkon rajaukseen.

Aina on hyvä varata jokaiselle hostille yksi ylimääräinen fyysinen verkkokortti hallintaverkon liikennettä varten ja pitää se rajattuna erillään virtuaalikoneiden liikenteestä.

### 3.6.1 Bonding

Verkon kahdentaminen on tärkeä osa minkä tahansa palvelimen pystyttämisessä. KVM-hostissa se tapahtuu samalla tavalla kuin muissakin Linux-palvelimissa, distribuution omia työkaluja käyttäen. Ubuntussa tarvitaan ifenslave-paketti, joka löytyy Ubuntun virallisista pakettilähteistä. Esimerkissä käytetään Ubuntu-distribuutiota.

Asennetaan ifenslave paketti.

```
sudo apt-get install ifenslave
```

Lisätään bonding-moduuli käynnistymään hostin käynnistyessä.



```
sudo sed -i '$ a\bonding' /etc/modules
```

Bondingissa voidaan käyttää seitsemää erilaista tilaa kahdentaa verkkoa. Voidaan esimerkiksi balansoida liikennettä kaikkien liitettyjen verkkokorttien kesken tasaisesti, tai jättää osa backup-tilaan. Valinta oikean tilan suhteen on hyvin ympäristökohtainen. (Ubuntu 2014)

Tehdään verkkoasetukset /etc/network/interfaces -tiedostoon jollain tekstieditorilla. Esimerkissä em2- ja em3-verkkokortit liitetään bond0-verkkosovittimeksi, jota voidaan käyttää vaikkapa Linux bridgen tai Open vSwitchin luomiseen myöhemmin. Sille voidaan myös haluttaessa antaa verkkoasetukset kuten normaalille verkkosovittimelle.

```
...
auto em2
iface em2 inet manual
    bond-master      bond0

auto em3
iface em3 inet manual
    bond-master      bond0

auto bond0
iface bond0 inet manual
    bond-mode        active-backup
    bond-miimon      100
    bond-slaves       none
...
```

Verkkoasetukset eroavat tietenkin esimerkistä monin tavoin ympäristöstä riippuen. Osa bonding-tiloista vaatii yhteensopivuutta ja konfigurointia verkkokytkimeltä.

### 3.6.2 Linux bridge

Linuxin verkkopinon vakiosilta on varsin pätevä ratkaisu, eikä sen pystyttäminen vaadi kuin hostin distribuutiokohtaisten verkkoasetusten muuttamista. Sillan avulla virtuaalikone saa jakaa hostin verkkokortin ja liittyy suoraan lähiverkkoon kuin kone olisi normaali fyysinen tietokone.

Linux bridge luodaan Ubuntussa tekstieditorilla /etc/network/interfaces -tiedostoon. Kahdennetun verkon verkkosovitinta voidaan käyttää bridge\_ports kohdassa.

```

...
auto br0
iface br0 inet static
    address      192.168.200.6
    netmask      255.255.248.0
    gateway      192.168.200.1
    dns-nameservers 172.31.14.130 172.31.14.131
    bridge_ports  bond0
    bridge_stp    on
    bridge_maxwait 0
    bridge_fd     0
...

```

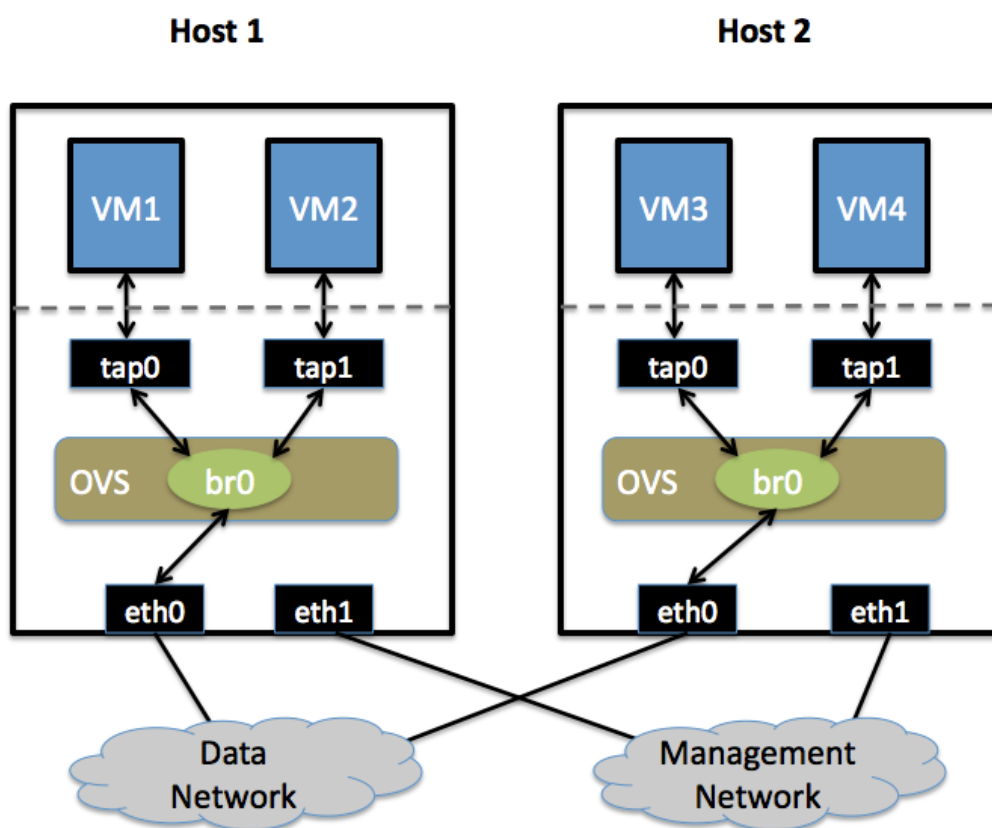
Verkkosovitinta br0 voidaan nyt käyttää virtuaalikoneiden verkkona.

Migraation toimivuuden vuoksi käytettävien verkkosovittimien nimet täytyy olla samat kaikilla hosteilla. (KVM c 2014)

### 3.6.3 Open vSwitch

Yksi vaihtoehto on käyttää Open vSwitch -virtuaalikytkintä Linuxin vakiosillan sijaan. Monet virtuaalikoneiden keskitetyt hallintajärjestelmät ovat adoptoineet sen vakio verkkoteknologiaan. Sen asennus komentoriviltä operoitavaan KVM-hostiin on myös helppoa.

Open vSwitch mahdollistaa monien edistyneiden ominaisuuksien käytön virtuaalisessa verkotuksessa. Esimerkiksi usean hostin yli yltävä yksityinen virtuaaliverkko ja virtuaalikonekohtainen verkonkäytön rajoitus, tyyliin VMwaren Distributed Switch, onnistuu Open vSwitchiä käyttäen. Se toimii myös suorana korvaajana Linuxin normaalille siltatilalle.



Kaavio 4. Open vSwitchin sijoittuminen virtualisointiympäristössä. (Open vSwitch 2014)

Vaikka Open vSwitch onkin varsin uusi tuote, libvirtistä löytyy jo integroitu tuki sille. Tarvitsee vain asentaa sen paketit, jotka löytyvät repositoriosta.

```
sudo apt-get install openvswitch-switch openvswitch-controller
```

Open vSwitchiin luodaan silta tilan verkkosovitin ja siihen liitetään halutut fyysiset verkkokortit.

```
ovs-vsctl add-br ovsbr0
```

```
ovs-vsctl add-port ovsbr0 bond0
```

Valitettavasti ohjelmat joilla ohjataan libvirttiä, kuten virt-install ja virt-manager, eivät ole vielä päivittyneet tukemaan Open vSwitchiä, vaan virtuaalikoneiden asetuksia joudutaan muokkaamaan manuaalisesti xml-tiedostosta, jotta ne osaisivat käyttää Open vSwitchiä.

```
virsh edit Virtuaalikone1
```

Voidaan joko tehdä uusi laite virtuaalikoneelle tai muokata olemassa olevaa.

```
...
<interface type='bridge'>
  <source bridge='ovsbr0' />
  <virtualport type='openvswitch' />
  <model type='virtio' />
</interface>
...
```

Open vSwitchin edistyneistä ominaisuuksista löytyy tietoa kotisivuilta osoitteesta <http://openvswitch.org/support/>.

Jos edistyneille toiminnoille ei ole tarvetta vaan virtuaalikoneiden rajoittamaton pääsy lähi-verkkoon riittää, ei Open vSwitchille ole tarvetta. Jos kuitenkin halutaan käyttää KVM:ää komentoriviltä, on vaikea nähdä syytä miksi ei käyttäisi Open vSwitchiä.

### 3.7 Virtuaalikoneiden hallinta

Virtuaalikoneita voidaan luoda, muokata ja poistaa KVM:ään libvirtin kautta. Ilman libvirttiä jouduttaisiin suorittamaan käskyjä suoraan qemu-kvm-ohjelmalle, ja ne käskyt ovat useita, jopa kymmeniä rivejä pitkiä.

#### 3.7.1 Virtuaalikoneen luominen

Virtuaalikoneiden luominen onnistuu parhaiten joko tuomalla asetukset xml-tiedostosta virsh-ohjelmalla, tai erinäisiä parametreja käyttäen virt-install-ohjelmalla.

Ensimmäisenä täytyy asentaa virtinst-paketti. Se löytyy ainakin Ubuntun pakettitietokannasta.

```
sudo apt-get install virtinst
```

Luodaan virtuaalikone. Parametrit ovat eri käyttöjärjestelmille hieman erilaiset mm. levyjen osalta, tai jos halutaan käyttää esimerkiksi SPICE:ä graafisena konsolina. Kaikista eri parametreista voi katsoa lisäinfoa ”man virt-install”-komennolla. Esimerkissä tehdään Ubuntu 13.10 virtuaalikone. Virt-install ei vielä tue Open vSwitchiä.

```
virt-install -v -n ubuntu2 --os-type=linux --os-variant=ubuntusaucy -r 1024 --network bridge=br0 --disk pool=lvm-vg0,size=16,bus=virtio,cache=none,sparse=false --graphics vnc,port=5902,listen=0.0.0.0 --noautoconsole -c ubuntu-13.10-server-amd64.iso
```

Virtuaalikone käynnistyy automaattisesti ja sen konsolinäkymään voidaan nyt yhdistää haluttu VNC-asiakasohjelmalla hostin IP-osoitteen ja määritellyn portin kautta ja viedä asennus loppuun.

Virtuaalikoneita voidaan luoda myös käyttäen xml-tiedostoa, johon asetukset on määritelty. Tämä tapa on hyödyllinen silloin, kun halutaan tehdä jo olemassa olevan virtuaalikoneen kaltainen uusi virtuaalikone. Olemassa olevan virtuaalikoneen asetukset voidaan viedä xml-tiedostoon, jonka jälkeen niitä voidaan muokata tarpeen mukaan.

```
virsh dumpxml ubuntu2 > ubuntu2.xml
```

Xml-tiedostoa voidaan muokata halutulla tekstinkäsittelyohjelmalla, esimerkiksi nano:lla. Tärkeimpiä muokkauksia on vaihtaa koneen nimi, UUID, MAC-osoite ja virtuaaliset levyt. UUID ja MAC-osoite osiot voidaan kokonaan poistaa, jolloin libvirt generoi ne automaattisesti uudelleen. Muita asetuksia voidaan muokata tarpeen mukaan, esimerkiksi RAM-muistin kokoa voidaan kasvattaa vaihtamalla numeroarvoa.

Virtuaalikone luodaan xml-tiedostosta komennolla:

```
virsh create ubuntu3.xml
```

Jos käytössä Linux-työpöytäkäyttöjärjestelmä, voidaan käyttää virt-viewer-ohjelmaa, joka yhdistää virtuaalikoneeseen hostin libvirt:n kautta.

### 3.7.2 Virtuaalikoneen muokkaaminen

Libvirttiin tehtyjen KVM-virtuaalikoneiden muokkaaminen onnistuu virsh- tai virt-manager-ohjelmien avulla. Molemmilla ohjelmilla käytännössä vain muokataan virtuaalikoneen asetus-tiedostoa.

Muokataan virtuaalikoneen asetuksia.

```
virsh edit ubuntu2
```

Virsh aukaisee virtuaalikoneen asetustiedoston käyttäjän vakiotekstinkäsittelyohjelmalla, esimerkiksi nano:lla, jossa muokkaukset voidaan tehdä.

Libvirtin xml-tiedostojen formaatti on selitetty syvällisemmin libvirtin kotisivuilla osoitteessa <http://libvirt.org/formatdomain.html>.

Asetukset päivittyvät vasta virtuaalikoneen uudelleenkäynnistyksen jälkeen.

Virt-manager tekee samat asiat kuin virsh mutta graafisesti.

### 3.7.3 Virtuaalikoneen poistaminen

Virtuaalikoneita voidaan poistaa libvirtistä yksinkertaisella komennolla.

Sammutetaan virtuaalikone.

```
virsh destroy ubuntu2
```

Poistetaan virtuaalikoneen asetustiedosto libvirtin muistista.

```
virsh undefine ubuntu2
```

Virtuaalikoneen asetusten poistaminen ei poista koneelle asetettua virtuaalista levyä, vaan se voidaan joko käyttää uudelleen tai poistaa. Virshissä on tuki näillekin poistoille, mutta se voidaan tehdä myös manuaalisesti poistamalla virtuaalikoneen levykuvatiedosto tai LVM-laite.

```
virsh vol-delete ubuntu2.img lvm-vg0
```

Samat toimenpiteet voidaan myös suorittaa graafisesti virt-manager ohjelmalla.

## 3.8 Migraatio

KVM-virtuaalikoneita voidaan migratoida toisille hosteille verraten helposti libvirtin kautta. Migraatiolla tarkoitetaan virtuaalikoneen siirtämistä toiselle fyysiselle palvelimelle suoritetta-

essa tai sammuksissa. Sen edellytyksenä on, että palvelinten kesken on jaettu jokin levyjärjestelmä, kuten iSCSI tai NFS, ja että molemmat fyysiset palvelimet ovat lähes identtisiä asetuksen ja laitteistonsa suhteen.

```
virsh migrate ubuntu1 qemu+ssh://192.168.200.9/system
```

Migratoidessa libvirt siirtää virtuaalikoneen asetustiedoston ja muistin toiselle hostille, jonka jälkeen se lopettaa virtuaalikoneen prosessin suorittamisen aiemmalla hostilla ja jatkaa sitä uudella. Virtuaalikoneelle toimenpide on lähes huomaamaton. Jotkin virtuaalikoneelle asetettut asetukset, kuten levyjen välimuisti, voivat estää migraation.

Migraatiosta ja sen toiminnasta löytyy kattavasti tietoa libvirtin kotisivuilta osoitteessa <http://libvirt.org/migration.html>.

### 3.9 Snapshot

Snapshotteja voidaan ottaa virtuaalikoneista KVM-ympäristössä. Jos käytössä on tiedostomuotoinen qcow2-levy, voidaan käyttää virshin snapshot-ominaisuuksia. LVM osioita käytettäessä voidaan käyttää LVM:n omia snapshot-toimintoja.

#### 3.9.1 Virsh

Virshin snapshot-ominaisuudet tukeutuvat QEMU:n varaan ja niinpä snapshotteja voidaan ottaa vain qcow2-levyjä käyttävistä virtuaalikoneista.

```
virsh snapshot-create-as ubuntu3 ubuntu3-snap "Testisnapshotti"
```

Virtuaalikoneen kaikista snapshoteista näkee tietoa ja päivämäärät.

```
virsh snapshot-list ubuntu3
```

Snapshotti voidaan tehdä joko sammutetusta tai käynnissä olevasta koneesta. Käynnissä olevan koneen tila ja ram-muisti tallennetaan levyjen lisäksi.

```
virsh snapshot-revert ubuntu3 ubuntu3-snap
```

Palautettaessa snapshottia virtuaalikone palaa siihen tilaan kuin se oli snapshottia otettaessa. Snapshottien poistaminen onnistuu myös virshillä. Jos snapshotilla on ”lapsi”-snapshotteja, poistettavan snapshotin data liitetään jäljellä oleviin.

```
virsh snapshot-delete ubuntu3 ubuntu3-snap
```

Kaiken kaikkiaan virshin snapshot-ominaisuus on todella kätevä, ja koska se käyttää qcow2-formaatin omaa snapshot-tukea, myös tilaa säästävä.

### 3.9.2 LVM

LVM-osioista voidaan ottaa snapshotteja LVM:n omilla työkaluilla. LVM:n snapshotit toimivat niin, että luodaan uusi erityinen snapshot volume, johon varsinaisesta volumesta tiedot kopioidaan vasta niiden muuttuessa. Tämä toteutus mahdollistaa snapshottien nopean luonnin, mutta on myös hyvin riippuvainen siitä, että snapshotti ja alkuperäinen levy sijaitsevat samalla hostilla. Luettaessa snapshot volumea luetaan käytännössä alkuperäistä volumea snapshot volumen läpi muutosten kera.

Snapshotin luonti tapahtuu lvcreate-komennolla. Snapshotille täytyy antaa koko, jona tulee olla suurempi kuin tulevien muutosten koko. Teoriassa koontarve voi kasvaa yhtä suureksi kuin alkuperäinen volume, mutta käytännössä yleensä tarvetta on vain murto-osalle koosta. Loppujen lopuksi koontarve on tilannekohtainen. Snapshotille on myös hyvä antaa asiaan sopiva nimi ”-n”-parametrilla.

```
lvcreate -L 1G -s -n ubuntu1.snap /dev/lvm-vg0/ubuntu1.img
```

Snapshottiin voi palata lvconvert-komennolla. Virtuaalikoneen tulee olla pois päältä, että toiminnon voi suorittaa. Snapshot poistetaan automaattisesti toiminnon suorittamisen jälkeen.

```
lvconvert --merge /dev/lvm-vg0/ubuntu1.snap
```

Snapshotin palauttaminen on peruuttamaton toimenpide.



### 3.10 Virtuaalikoneen tuonti toisesta ympäristöstä

Jos halutaan tuoda virtuaalikone KVM-ympäristöön jostain toisesta ympäristöstä, kuten VMwaren ESXi:stä, joudutaan käytännössä luomaan samoilla asetuksilla oleva virtuaalikone KVM-ympäristöön ja konvertoimaan virtuaalikoneen levy qemu-img-ohjelmalla.

```
qemu-img convert vmware_koneen_levy-flat.vmdk -O qcow2 vm1.qcow2
```

Voidaan myös luoda LVM block device, jos halutaan. LVM osio täytyy olla etukäteen tehtynä ja samankokoinen tai suurempi kuin tuotava levy.

```
qemu-img convert vmware_koneen_levy-flat.vmdk -O raw temp2.raw
```

```
dd if=temp2.raw of=/dev/lvm-vg0/vm2
```

Kopioinnin onnistuttua väliaikaisen raw-tiedoston voi poistaa.

Luotua levyä voidaan käyttää KVM-virtuaalikoneessa ja se sisältää samat tiedot kuin aiemmassa ympäristössä.

### 3.11 High Availability

KVM:n High Availability klusteri on mahdollista ilman kaiken kattavaa järjestelmää, mutta sen pystyttäminen on vaikeaa ja vaatii useita erinäisiä ohjelmia ja niiden konfigurointia manuaalisesti. Osassa distribuutioissa osat ominaisuuksista eivät edes toimi. Käytännön testeissä todettiin tällaisen manuaalisen rakennelman olevan liian vaivalloinen pystyttää.

Jos High Availability on ominaisuus jota tarvitaan, kannattaa pystyttää jokin valmiiksi paketoitu kaiken kattava järjestelmä, kuten vaikka Proxmox VE.

## 4 PROXMOX VE

Proxmox VE eli Proxmox Virtual Environment on yksi suosituimmista KVM:n ympärille rakennetuista ilmaisista järjestelmistä. Se on avoimen lähdekoodin sovellus ja se lisensoitu GNU AGPL v3 mukaan. Proxmox VE on pakattu omaksi distribuutiokseen, joten sitä voidaan pitää bare-metal-virtualisointiratkaisuna. Käytännössä se on vain Debian-pohjainen Linux-distribuutio, jossa on muokattu kernel ja hallintatyökalut valmiiksi asennettuina. Proxmoxin paketit onkin mahdollista asentaa manuaalisesti Debianin päälle, joskin syytä tuolle toimenpiteelle on vaikea ymmärtää.

Proxmox VE:tä ylläpitää ja kehittää itävaltalainen Proxmox Server Solutions GmbH. Heillä on myös toinen tuote nimeltä Proxmox Mail Gateway. (Proxmox a 2014)

Proxmoxin käyttöliittymä on WebUI ja virtuaalikoneiden VNC konsoli-ikkuna on toteutettu Java:lla, joten sen käyttäminen onnistuu lähes millä tahansa alustalla.

### 4.1 Subscription

Proxmox VE:tä voi ladata ja käyttää ilmaiseksi, mutta siitä löytyy myös maksullisia lisenssejä, joissa tarjotaan Enterprise-tason käyttäjätukea ja repositoryä.

COMMUNITY	BASIC	STANDARD	PREMIUM
€ 4,16 / CPU & month	€ 16,58 / CPU & month	€ 33,17 / CPU & month	€ 66,33 / CPU & month
<a href="#">Buy now</a>	<a href="#">Buy now</a>	<a href="#">Buy now</a>	<a href="#">Buy now</a>
✓ Access to Enterprise Repository	✓ Access to Enterprise Repository	✓ Access to Enterprise Repository	✓ Access to Enterprise Repository
✓ Stable software updates	✓ Stable software updates	✓ Stable software updates	✓ Stable software updates
✓ Support via community forum	✓ Support via Customer Portal	✓ Support via Customer Portal	✓ Support via Customer Portal
	✓ 3 support tickets/year	✓ 10 support tickets/year	✓ Unlimited support tickets
	✓ Response time: 1 business day	✓ Response time: 1 business day	✓ Response time: 1 business day
	✓ Get additional support tickets	✓ Remote login via SSH	✓ Remote login via SSH
		✓ Get additional support tickets	

Kuva 2. Proxmox VE:n maksulliset lisenssimallit. (Proxmox e 2014)

Tarjolla on neljän eri tason paketteja, joiden hinnat vaihtelevat alle viidestä eurosta fyysinen/proessori/kuukausi, mikä tarjoaa käytännössä vain repositorion, aina kaiken kattavaan yli 65 euron fyysinen/proessori/kuukausipakettiin asti.

## 4.2 Teknologiat

Proxmox VE käyttää pitkälti samoja teknologioita kuin komentoriviltä toimiva KVM libvirt-yhdistelmä, mutta Proxmox VE ei käytä libvirtä rajapintanaan. Kehitystiimi on luonut oman järjestelmänsä hoitamaan samoja asioita, joita libvirt hoitaa, ja tehnyt sen päälle kätevän web-käyttöliittymän.

Verkkoteknologioina ovat tuettuna bonding, bridge ja NAT. 3.2-versiosta lähtien myös Open vSwitch on tuettuna, mutta se on epävakaa ja testivaiheessa.

Tallennusteknologioita on tuettuna iSCSI, LVM, NFS, GlusterFS ja 3.2 versiosta lähtien Ceph RBD. Suosittu kokoonpano on käyttää iSCSI levyjä LVM:n alustana, josta näytetään loogisia levyjä virtuaalikoneille. (Proxmox b 2014)

Graafisesta konsolista vastaa joko VNC tai SPICE.

KVM-virtualisoinnin lisäksi Proxmox VE:llä voidaan luoda OpenVZ konteinereja ja hallita niitä kuin virtuaalikoneita. OpenVZ on yleinen teknologia virtuaalipalvelimia myyvien palveluntarjoajien keskuudessa.

## 4.3 Laitteistovaatimukset

Proxmoxin sivuilla listaamat minimivaatimukseksi hosteille ovat varsin vaatimattomat: 64-bittinen prosessori (AMD64 tai EMT64), Intel VT tai AMD-V tuki prosessorilta ja emolevyltä, 1 Gigatavu RAM-muistia, kiintolevy ja verkkokortti.

Suosittelut laitteisto taas on huomattavasti vaativampi mutta hyvin normaali palvelinkäytössä

- Usean prosessorin palvelin, jossa joka prosessorilla useita suoritintimejä

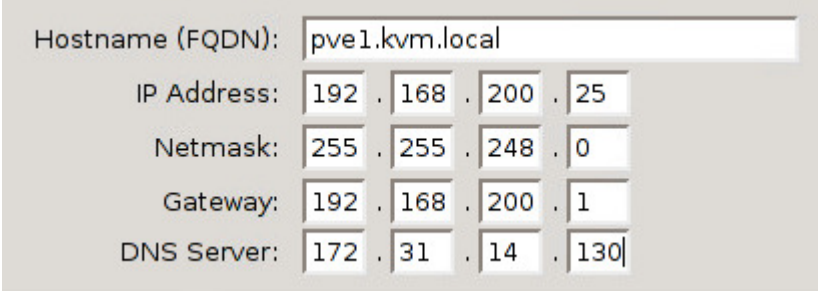
- 64-bittinen prosessori (AMD64 tai EMT64)
- Intel VT tai AMD-V tuki prosessorilta ja emolevyltä
- 8 Gigatavua RAM-muistia, mutta mitä enemmän, sen parempi
- Laitteistopohjainen RAID-tallennusjärjestelmä nopeilla levyillä tai jokin muu vastaava tallennusratkaisu
- Kaksi tai enemmän Gigabitin verkkokortteja riippuen tiedonsiirron tarpeesta
- Fencing-laitteisto, jos halutaan High Availability ominaisuudet.

(Proxmox c 2014)

#### 4.4 Asennusprosessi

Koska Proxmox VE om ns. baremetal ratkaisu, se asennetaan tyhjälle palvelimelle, kuten mikä tahansa muukin Linux-distribuutio, levyiltä tai USB-tikulta. Uusimman version ISO-levykuvatiedostot on saatavilla Proxmox VE:n kotisivuilta osoitteessa <http://www.proxmox.com/downloads> .

Proxmoxin asennusprosessi on hyvin selkeä ja suoraviivainen. Asennuksen aikana määritellään IP-osoite ja muut verkkoasetukset, hostname ja pääkäyttäjän salasana. Asennuksen jälkeen voidaan käyttää ssh:ta komentojen suorittamiseen.



Hostname (FQDN):	pve1.kvm.local			
IP Address:	192	168	200	25
Netmask:	255	255	248	0
Gateway:	192	168	200	1
DNS Server:	172	31	14	130

Kuva 3. Proxmox VE:n asennuksen verkkoasetusten määrittäminen.

Web-käyttöliittymä löytyy vakiona jokaiselta asennetulta hostilta portista 8006 käyttäen HTTPS-protokollaa, esimerkiksi: <https://192.168.200.11:8006> .

#### 4.4.1 Klusterin luominen

Usean hostin klusterin luomiseen ei ole vielä graafista mahdollisuutta, vaan se joudutaan tekemään komentorivin puolelta, esimerkiksi SSH-yhteyttä käyttäen.

Tehdään klusteri yhdellä hosteista.

```
pvecm create proxmox-cluster
```

Lopuilla hosteista ajetaan komento, joka lisää ne tehtyyn klusteriin (IP koneesta jolla tehtiin klusteri).

```
pvecm add 192.168.200.11
```

Klusterin luomisen ja hostien siihen liittämisen jälkeen web-käyttöliittymässä näkyy useita hosteja.

#### 4.4.2 Rekisteröintimuistutuksen poistaminen

Jos halutaan eroon kirjautumisen ohessa tulevasta viestistä, joka huomauttaa ettei Proxmox VE asennusta ole rekisteröity, voidaan jokaisella hostilla ajaa pieni muokkaava päivitys, joka muokkaa `/usr/share/pve-manager/ext4/pvmanagerlib.js` -tiedostoa. Päivitys on jonkun tuntemattoman käyttäjän luoma eikä virallisesti tuettu. Se toimii ainakin testaushetkellä Proxmox VE 3.1-versiossa.

```
wget http://www.tolaris.com/blog/wp-content/uploads/2013/08/no_subscription_popup_pvmanagerlib.js-3.1.patch -O - | patch -p0
```

#### 4.4.3 Repositorion vaihtaminen

Vakiona Proxmox VE:ssä on asetuksissa määritettynä maksullinen enterprise-repositorio. Sen voi vaihtaa maksuttomaan versioon tai vain poistaa käytöstä.

Poistetaan Enterprise-repositorio käytöstä lisäämällä kommentimerkki APT:n asetustiedostoon, jossa repositorio on määritetty, tai vain poistamalla tiedosto.

```
rm /etc/apt/sources.list.d/pve-enterprise.list
```

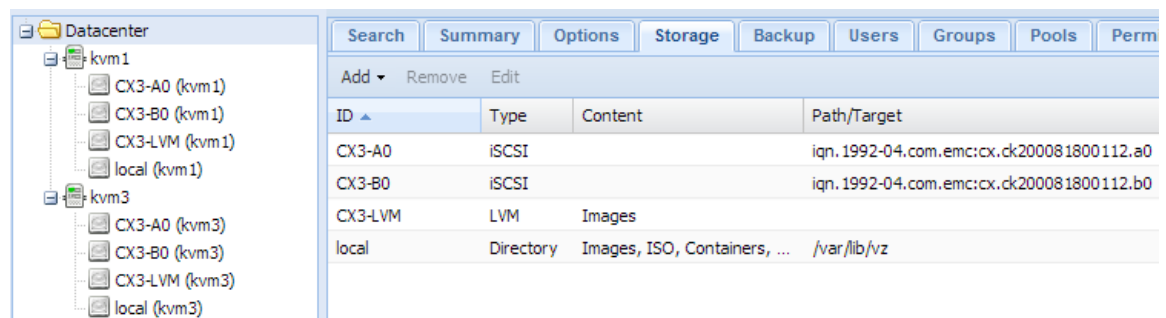
Ilmaisen repositorion voi ottaa käyttöön lisäämällä repositorion apt:n asetuksiin, esimerkiksi ”/etc/apt/sources.list”-tiedostoon.

```
deb http://download.proxmox.com/debian wheezy pve-no-subscription
```

Kannattaa ottaa huomioon, että ilmainen repositorio on käytännössä testauskäytössä ja sen paketit eivät ole välttämättä yhtä vakaita kuin enterprise-repositorion vastaavat.

## 4.5 Tallennus

Storagea määritellään Proxmox VE:ssä koko klusterille, eikä yksittäisille hosteille. Storageja voidaan kuitenkin rajata käyttöön vain osalle hosteista niin haluttaessa.



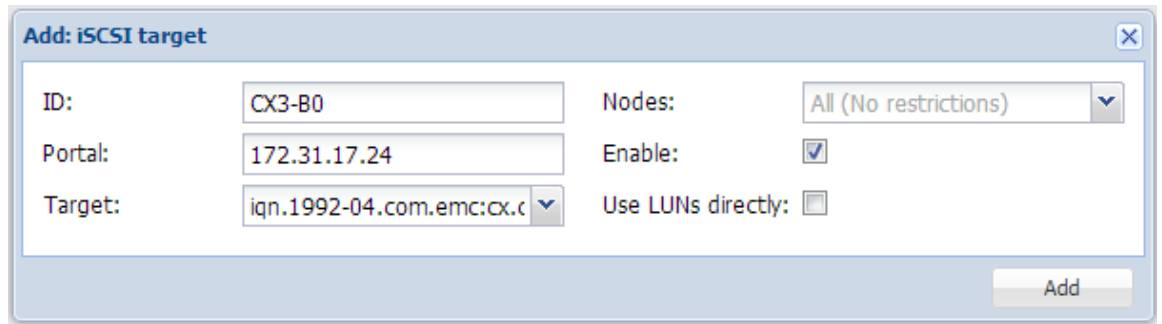
Kuva 4. Storage-näkymä Proxmox VE:ssä.

Proxmox VE tukee NFS-, iSCSI-, LVM-, GlusterFS- ja RDB-teknologioita tallennukseen. Manuaalisesti voidaan asentaa toki muitakin ratkaisuja tai käyttää paikallisia kansioita tallennukseen.

### 4.5.1 iSCSI + LVM

Proxmox VE sisältää valmiiksi asennettuna iSCSI-initiaattorin. Initiaattorin osoitteen voi halutessaan vaihtaa ”/etc/iscsi/initiatorname.iscsi”-tiedostoon. Muutoksen jälkeen hosti täytyy käynnistää uudelleen.

Jos iSCSI-levyjärjestelmään on luotu LUN:it ja hosteilla on oikeus käyttää niitä, on iSCSI targettien liittäminen Proxmox VE:seen varsin helppoa. Targettia lisättäessä täytyy määrittellä sille jonkin uniikki ID ja portaalin IP-osoite, jonka jälkeen portaalista skannatun targetin IQN-osoitteen voi valita.

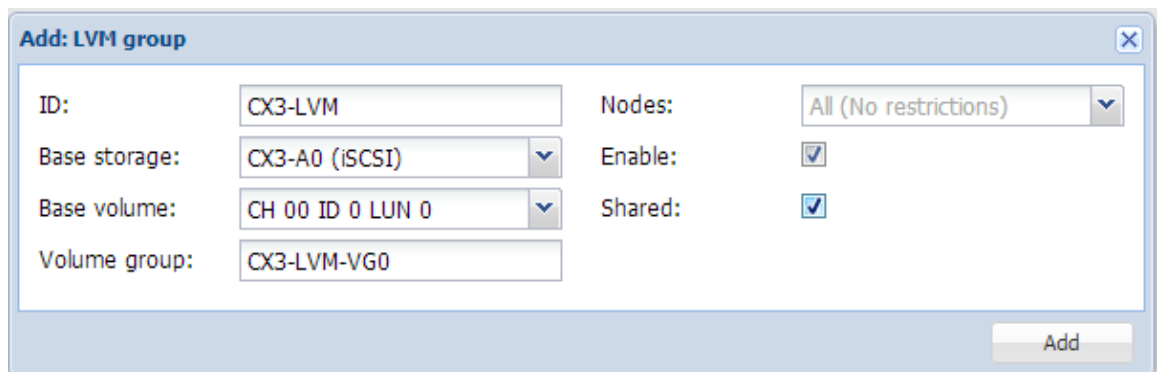


ID:	CX3-B0	Nodes:	All (No restrictions)
Portal:	172.31.17.24	Enable:	<input checked="" type="checkbox"/>
Target:	iqn.1992-04.com.emc.cx.c	Use LUNs directly:	<input type="checkbox"/>

Add

Kuva 5. iSCSI-targetin lisääminen Proxmox VE:ssä.

iSCSI:n lisäämisen jälkeen voidaan luoda LVM volume group sen päälle. LVM-storagelle määritellään uniikki ID ja valitaan haluttu alusta, tässä tapauksessa luotu iSCSI-storage. Storagen valinnan jälkeen voi valita haluamansa LUN:in, johon LVM volume group luodaan. LUN:it näkyvät vain jos hostilla on oikeudet käyttää sitä. Volume groupille pitää antaa myös uniikki ID.



ID:	CX3-LVM	Nodes:	All (No restrictions)
Base storage:	CX3-A0 (iSCSI)	Enable:	<input checked="" type="checkbox"/>
Base volume:	CH 00 ID 0 LUN 0	Shared:	<input checked="" type="checkbox"/>
Volume group:	CX3-LVM-VG0		

Add

Kuva 6. LVM-storagen luonti Proxmox VE:ssä.

LVM-storagen luonnin jälkeen virtuaalikoneita voidaan luoda sen päälle. LVM:ää käytettäessä jokaiselle virtuaalikoneelle luodaan looginen osio, joten sinne voidaan tallentaa muuta kuin virtuaalikoneiden levyjä

#### 4.5.2 NFS

NFS on helpoin tapa käyttää jaettua storagea Proxmox VE:n kanssa. Siten voidaan tallentaa tiedostomuotoiset virtuaalikoneiden levyt, OpenVZ konteinerit ja iso-levykuvat.

Storagelle täytyy antaa uniikki ID ja IP-osoite tai DNS-nimi, jossa NFS-jako sijaitsee. Proxmox VE skannaa NFS-palvelimen jaot ja niistä voidaan valita haluttu jako. Voidaan myös määrittää, mitä kaikkea Proxmox VE voi tallentaa kyseessä olevaan storageen. Tarvittaessa voidaan luoda esimerkiksi omat storaget erikseen iso-levykuville ja virtuaalikoneiden levyille.

Kuva 7. NFS-storagen luonti Proxmox VE:ssä.

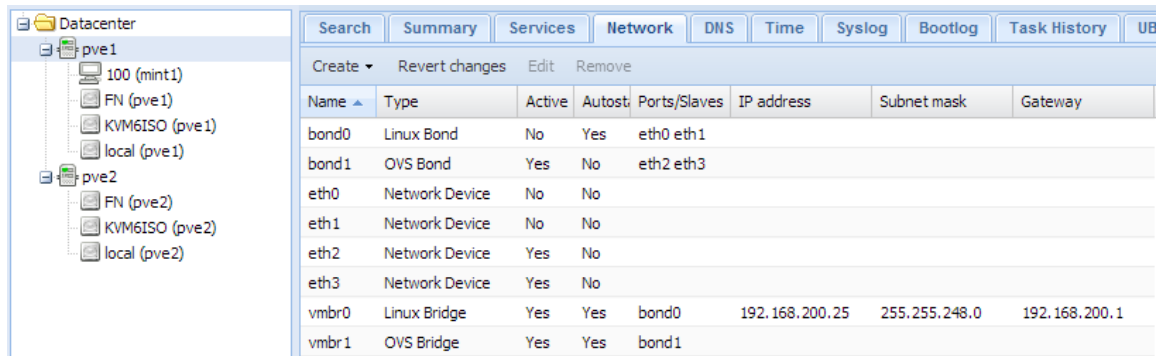
Luotaessa virtuaalikoneita NFS-storageen virtuaalikoneiden levyt luodaan qcow2-, raw- tai vmdk-muodossa.

#### 4.6 Verkko

Proxmox VE:n verkko-ominaisuudet ovat hyvin samanlaiset kuin normaalit Linux-palvelimesta ajettavan KVM-hostin. Voidaan käyttää joko Linuxin vakio bondingia ja siltaa, tai voidaan käyttää Open vSwitchiä. Open vSwitch -tuki on lisätty Proxmox VE 3.2 -versiossa.

Verkkosovittimien nimet on määritetty Proxmox VE:ssä käyttämään samaa kaavaa. Fyysisten verkkokorttien nimet ovat muodossa eth0, jossa numero on väliltä 0-99. Bonding-tilan verkkosovittimien nimet ovat muotoa bond0, jossa numero on väliltä 0-99. Silta-tilan verkkosovittimien nimet ovat muotoa vbr0, jossa numero on väliltä 0-99.





Name	Type	Active	Autostart	Ports/Slaves	IP address	Subnet mask	Gateway
bond0	Linux Bond	No	Yes	eth0 eth1			
bond1	OVS Bond	Yes	No	eth2 eth3			
eth0	Network Device	No	No				
eth1	Network Device	No	No				
eth2	Network Device	Yes	No				
eth3	Network Device	Yes	No				
vmbr0	Linux Bridge	Yes	Yes	bond0	192.168.200.25	255.255.248.0	192.168.200.1
vmbr1	OVS Bridge	Yes	Yes	bond1			

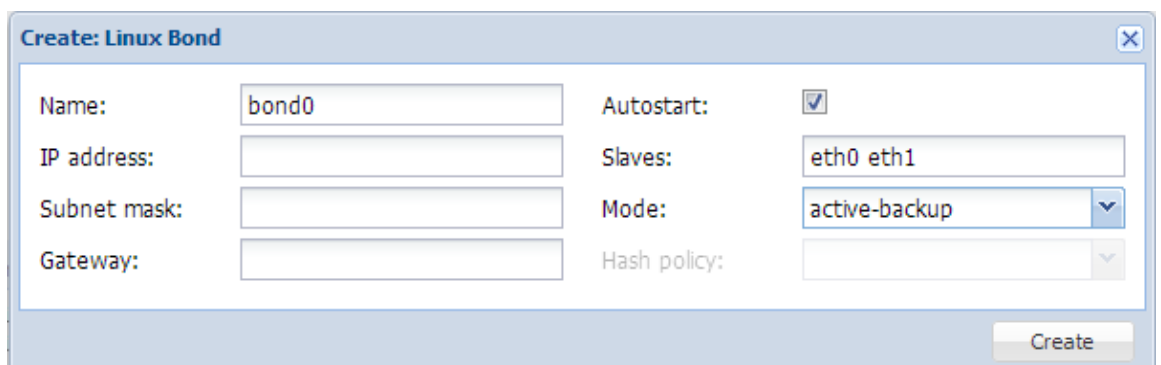
Kuva 8. Proxmox VE:n verkkonäkymä.

Silta-tilan verkkoon liitettäville virtuaalikoneille luodaan jokaiselle oma verkkosovitin, joka liittyy siltaan. Niiden nimet ovat muotoa ”tap104i0”, jossa ensimmäinen numero on virtuaalikoneen ID ja toinen on virtuaalikoneen verkkosovittimen numero.

#### 4.6.1 Linux Bonding ja Bridge

Vakiona Proxmox VE luo Linuxin siltatilassa toimivan verkkosovittimen vmbr0, joka käyttää ensimmäistä hostista löytyvää verkkokorttia. Siltänsä se on toimiva ja siihen voidaan liittää virtuaalikoneita, mutta luultavasti halutaan luoda kahdennettu verkkoja ja rajata levyliikenne, hallintaliikenne ja virtuaalikoneiden liikenne eri verkkoihin.

Verkon kahdennus on toteutettu Linuxin bonding-moduulilla. Bondingissa voidaan käyttää kaikkia sen tukemia tiloja, kuten balance-rr, balance-tlb tai active-backup. Kahdennelle verkolle täytyy tietenkin määrittää kaksi tai useampi fyysistä verkkokorttia. IP-asetuksia tehtäville bond-verkkosovittimelle ei tarvitse määrittää.



**Create: Linux Bond**

Name:  Autostart: ☒

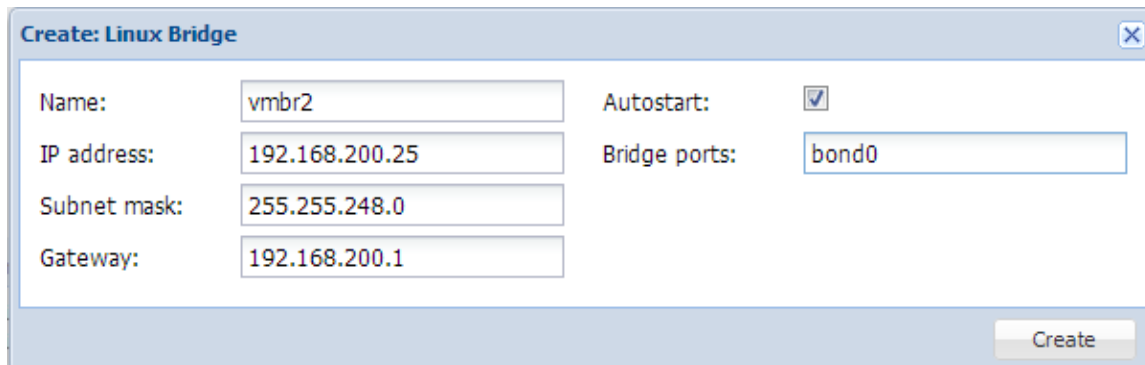
IP address:  Slaves:

Subnet mask:  Mode:

Gateway:  Hash policy:

Kuva 9. Linux Bond -verkkosovittimen luonti Proxmox VE:ssä.

Tehtyä bonding-tilan verkkosovitinta voidaan käyttää Linux bridgen alla, jolloin virtuaalikoneiden liikenne on kahdennettu. Sillalle voidaan myös antaa IP-asetukset, jolloin sitä voidaan käyttää hallintatoimenpiteisiin eikä tarvita erillistä verkkokorttia.



Name:	vmbr2	Autostart:	<input checked="" type="checkbox"/>
IP address:	192.168.200.25	Bridge ports:	bond0
Subnet mask:	255.255.248.0		
Gateway:	192.168.200.1		

Create

Kuva 10. Linux Bridge -verkkosovittimen luonti Proxmox VE:ssä.

Tehtyä silta-tilan verkkosovitinta voidaan käyttää virtuaalikoneiden verkkona, jolloin ne toimivat kuin olisivat fyysisiä koneita verkkoon liitettyinä.

Tehtyjä asetusmuutoksia voi tarkastella hallintapaneelin alaosasta. Uudet asetukset tulevat voimaan vasta hostin uudelleenkäynnistyksen jälkeen.

#### 4.6.2 Open vSwitch bonding ja bridge

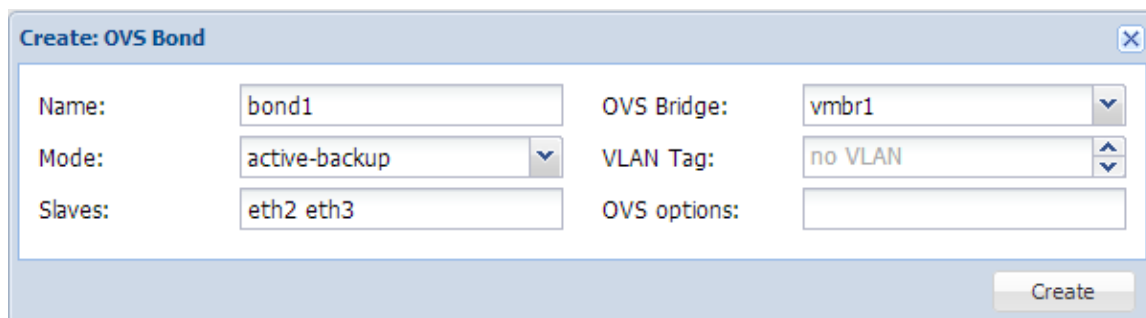
Proxmox VE:n 10.3.2014 julkaistussa 3.2-versiossa lisätty Open vSwitch -tuki on beta-vaiheessa, eikä sen käyttöä suositella tuotantoympäristössä. (Proxmox d 2014)

Vaikka Proxmox VE:n hallintapaneelistä löytyy tuki Open vSwitchille, sen käyttäminen vaatii pakettien asentamista hostille komentoriviltä.

```
aptitude install openvswitch-switch
```

Asennuksen jälkeen ensimmäinen vaihe on luoda OVS Bridge -verkkosovitin Proxmox VE:n hallintapaneelin verkkonäkymästä. Sen asetuksiin ei tarvitse määritellä mitään muita asetuksia kuin verkkosovittimen nimi.

Kahdennettua verkkoa varten luodaan OVS Bond -verkkosovitin, jonka asetuksiin määritellään nimen lisäksi bonding-tila, liitettävät fyysiset verkkokortit ja käytettävä OVS Bridge.



**Create: OVS Bond**

Name:  OVS Bridge:

Mode:  VLAN Tag:

Slaves:  OVS options:

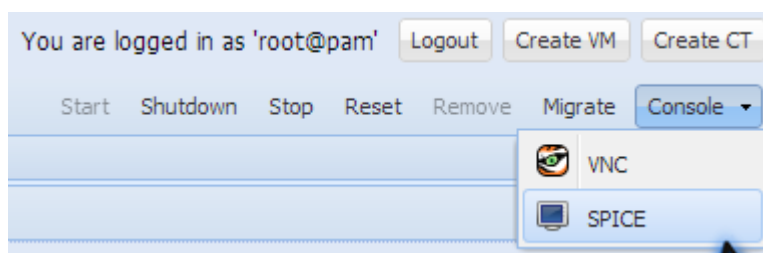
Kuva 11. Open vSwitch Bond -verkkosovittimen luonti Proxmox VE:ssä.

Tehtyä silta-tilan verkkosovitinta voidaan käyttää virtuaalikoneiden verkkona, jolloin ne toimivat kuin olisivat fyysisiä koneita verkkoon liitettyinä.

Tehtyjä asetusmuutoksia voi tarkastella hallintapaneelin alaosasta. Uudet asetukset tulevat voimaan vasta hostin uudelleenkäynnistyksen jälkeen.

#### 4.7 Virtuaalikoneiden hallinta

Virtuaalikoneita hallitaan Proxmox VE:ssä webui:n kautta. Paneelin oikeaan yläkulmaan on järjestetty kaikki keskeiset virtuaalikoneen hallintatoiminnot. Samat toiminnot löytyvät myös hiiren valikosta.



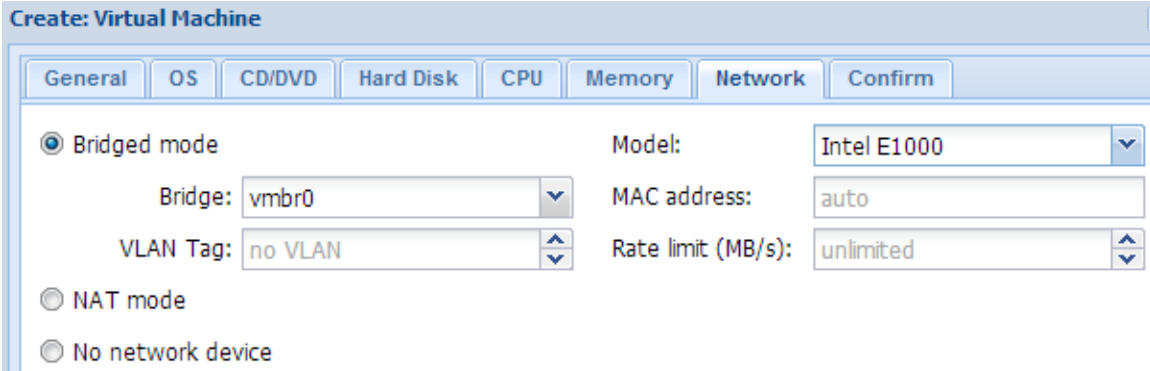
Kuva 12. Proxmox VE:n virtuaalikoneen hallintoiminnot.

Konsoli-ikkunan avauksessa voidaan valita VNC- tai SPICE-teknologian käyttö. VNC-konsoli on toteutettu Java-ohjelmointikielellä, joten sen pitäisi toimia alustalla kuin alustalla. SPICE:n käyttö vaatii työasemalta SPICE:ä tukevan virt-viewer-asiakasohjelman, joka löytyy Linux- ja Windows-käyttöjärjestelmille. Sen voi ladata osoitteesta <http://virt-manager.org/download/>.

#### 4.7.1 Virtuaalikoneen luonti

Proxmox VE:n hallintapaneelin yläkulmasta löytyy napit KVM-virtuaalikoneiden ja OpenVZ-konteinereiden luomiseen.

Virtuaalikoneen luonti on varsin itsestään selvää ja vaivatonta. Valitaan haluttu nimi, käyttöjärjestelmän tyyppi, käytettävä iso-asennuskuva, levy, muisti ja verkko. Levyjä ja verkkoja voi luonnissa tehdä vain yhdet, mutta niitä voi lisätä myöhemmin lisää muokkaamalla virtuaalikonetta. Levyjä voi tehdä vain asiaankuuluviin storageihin.



**Create: Virtual Machine**

General OS CD/DVD Hard Disk CPU Memory **Network** Confirm

☒ Bridged mode      Model: Intel E1000

Bridge: vmbr0      MAC address: auto

VLAN Tag: no VLAN      Rate limit (MB/s): unlimited

☐ NAT mode

☐ No network device

Kuva 13. Proxmox VE:n virtuaalikoneen luonnin verkkoasetukset.








Vakiona Proxmox VE ehdottaa virtuaalikoneen levyn tyyppiä IDE ja verkkosovittimen tyyppiä Intel E1000, mutta riippuen virtuaalikoneen tulevasta käyttöjärjestelmästä voidaan haluta käyttää jotain muuta vaihtoehtoa. Linux-virtuaalikoneita luotaessa on yleisesti hyvä käyttää VirtIO:ta. Paravirtualisoidun VirtIO:n käyttö vähentää kuormaa hostilla, kun ei jouduta emuloimaan fyysisiä laitteita.

Luotu virtuaalikone ei käynnisty automaattisesti. Virtuaalikoneen käyttöjärjestelmän asennuksen voi aloittaa konsoli-ikkunan kautta.

#### 4.7.2 Virtuaalikoneen muokkaaminen

Proxmox VE:hen tehtyjä virtuaalikoneita voidaan muokata monilta osin. Virtuaaliset laitteet ja ns. BIOS-asetukset on jaettu eri osioikseen. Suoritinta, RAM-muistia, levyjä ja verkko-

sovittimia voidaan lisätä samaan tyyliin kuin virtuaalikonetta luodessa. Näytön rajapintaa voidaan myös vaihtaa VNC:n, SPICE:n ja muutaman muun vaihtoehdon kesken.

Add ▾ Remove Edit Resize disk Move disk Disk Throttle	
 Keyboard Layout	Default
 Memory	1.00GB
 Processors	1
 Display	SPICE (qxl)
 CD/DVD Drive (ide2)	none,media=cdrom
 Hard Disk (virtio0)	ide-lvm:vm-100-disk-1,size=8G
 Network Device (net0)	e1000=CA:2C:4F:B1:38:B5,bridge=vbr0

Kuva 14. Proxmox VE:n virtuaalikoneen laitteiden hallintapaneeli.

Yksi varsin vakava puute virtuaalikoneiden muokkauksessa on, että olemassa olevia levyjä ei voi uusiokäyttää käyttöliittymän kautta. Vanhoja levyjä voi lisätä manuaalisesti vain muokkaamalla virtuaalikoneiden asetustiedostoja tai luomalla uuden samanlaisen levyn, jonka tilalle vanhan siirtää.

Asetuksista voidaan säätää samoja asetuksia Proxmox VE:n omien asetusten lisäksi asetuksia, joita fyysisen koneen BIOS-asetuksissa säädettäisiin, kuten käynnistysjärjestystä.

Edit	
Name	roskaloota-www
Start at boot	Yes
Start/Shutdown order	order=any
OS Type	Linux 3.X/2.6 Kernel (i26)
Boot order	Disk 'virtio0', CD-ROM, Network
Use tablet for pointer	Yes
ACPI support	Yes
SCSI Controller Type	Default (LSI 53C895A)
KVM hardware virtualization	Yes
CPU units	1000
Freeze CPU at startup	No
Use local time for RTC	No
RTC start date	now

Kuva 15. Proxmox VE:n virtuaalikoneen asetukset

### 4.7.3 Virtuaalikoneen poistaminen

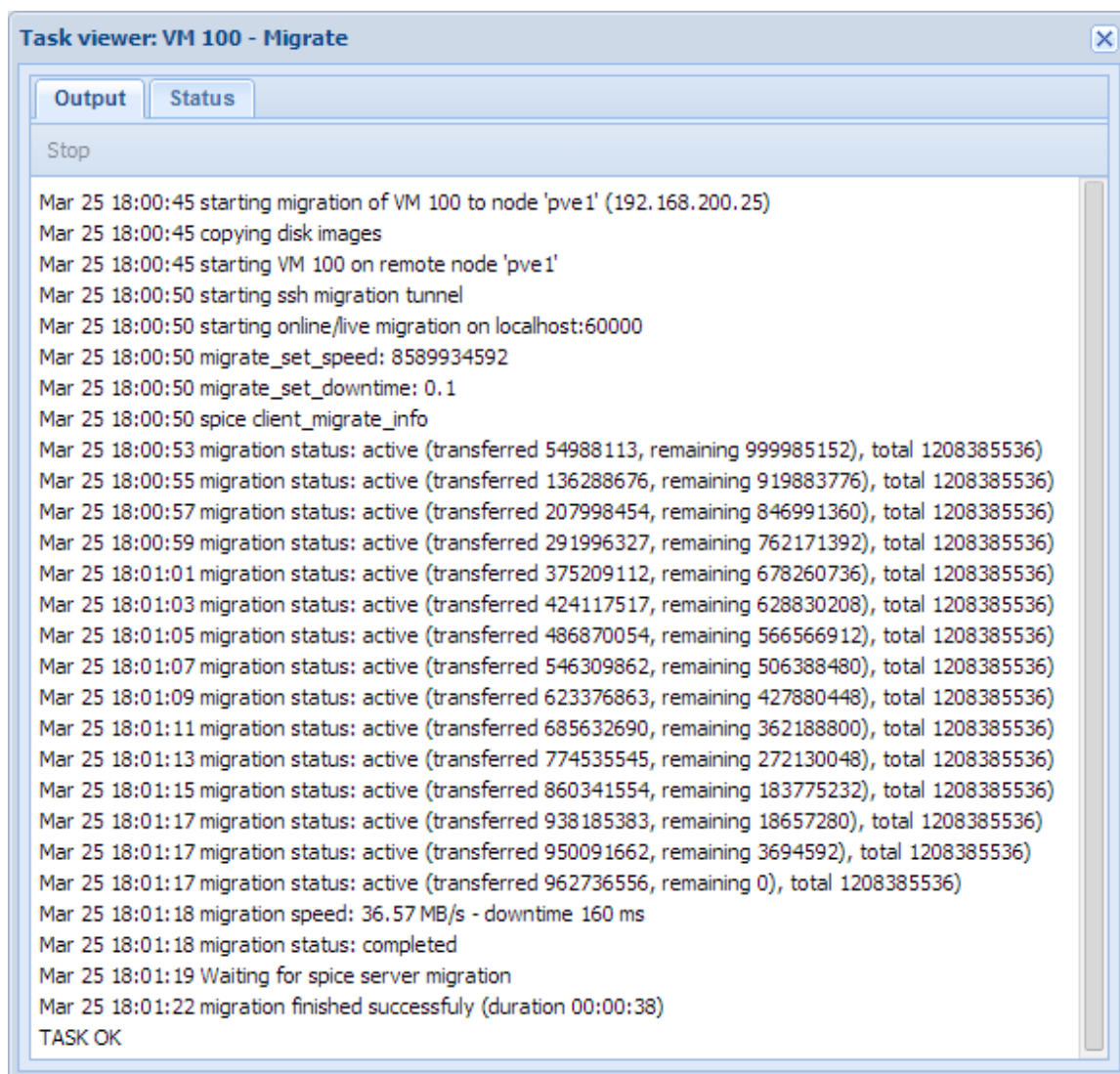
Virtuaalikoneita voi poistaa Proxmox VE:ssä varsin helposti kahden napin painalluksella. Virtuaalikoneen poistaminen poistaa kaiken siihen liittyvän datan. Tämä tarkoittaa levyjä, siltaan luotuja verkkosovittimia ja asetustiedostoja. Virtuaalikoneen poistaminen vapauttaa sen ID:n muuhun käyttöön.

Poistotoimenpide on peruuttamaton ja virtuaalikone voidaan palauttaa ainoastaan varmuuskopiosta, jos sellainen on tehty.

## 4.8 Migraatio

Proxmox VE:ssä voidaan migratoida virtuaalikoneita hostien välillä suoritettaessa tai samuksissa. Vaatimukset migratoinnille ovat jaettu storage ja saman nimisten verkkolaitteiden löytyminen. Live-migraatio vaatii myös, että virtuaalikoneen levyn välimuisti-tilaksi on valittu ”none”. yhteensopivat prosessorit hosteilla.

Migraatiossa virtuaalikoneen asetustiedosto siirtyy fyysiseltä hostilta toiselle ja live-migraatiossa siirretään myös virtuaalikoneen RAM-muisti, virtuaalinen verkkolaite ja mahdollinen SPICE-istunto.



Kuva 16. Proxmox VE:n live-migraation tehtävälogi.

Testatussa tilanteessa siirrettiin yhden gigatavun RAM-muistilla varustettu virtuaalikone, jonka levy sijaitsi NFS-storagessa ja verkkolaitteena käytettiin OVS-siltaa.

Käyttäjälle live-migraatio on lähes huomaamaton. Vain SPICE-istunnon siirtyessä käyttäjään toista hostia ruutu välähti mustana hetkellisesti. Migraation aikana testikoneessa ajettun ping-ohjelman paketeista kaikki saapuivat perille onnistuneesti. Proxmox VE:n tehtävälogiin mukaan virtuaalikone oli tavoittamattomissa 160 millisekuntia.

```

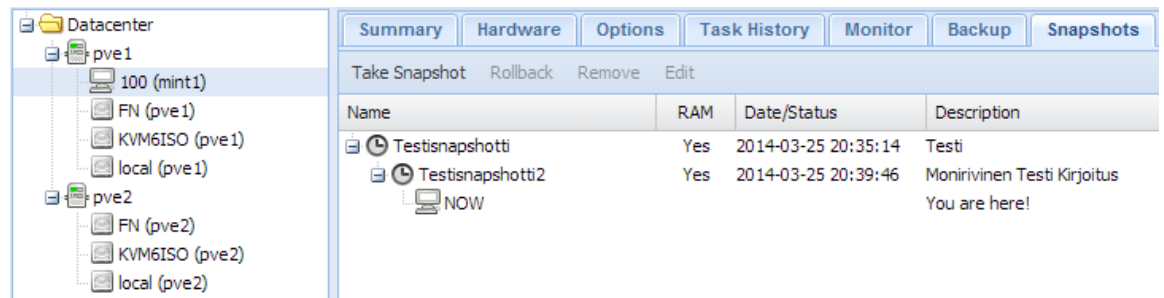
--- 8.8.8.8 ping statistics ---
56 packets transmitted, 56 received, 0% packet loss, time 55059ms
rtt min/avg/max/mdev = 29.169/29.823/33.501/0.841 ms

```

Kuva 17. Virtuaalikoneen ping-ohjelman tilastit migraation aikana kohteeseen 8.8.8.8.

#### 4.9 Snapshot ja Backup

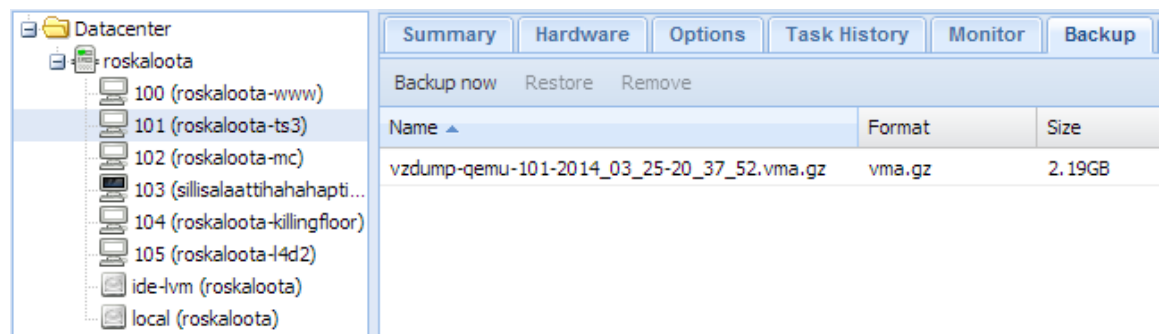
Proxmox VE:ssä voidaan ottaa snapshotteja tiedosto-muodossa olevista virtuaalikoneista. Snapshot toiminto käyttää qcow2-formaatin omia snapshot toimintoja, joten se säilöo ne samaan tiedostoon kuin alkuperäinen levy.



Kuva 18. Snapshot-näkymä Proxmox VE:ssä.

Virtuaalikoneesta tehtävään snapshottiin voidaan sisällyttää RAM-muisti, jos snapshot tehdään virtuaalikonetta suoritettaessa. Snapshottiin voidaan palata vaikka suorituksen aikana.

LVM-muodossa olevista virtuaalikoneista ei voi ottaa snapshotteja samalla toiminnolla, mutta niistä voidaan ottaa varmuuskopioita, jotka voidaan ottaa snapshot muodossa. Varmuuskopiot toimivat huomattavasti hitaammin, kuin snapshotit. Varmuuskopioita voi ottaa myös tiedostomuotoisista virtuaalikoneista.



Kuva 19. Backup-näkymä Proxmox VE:ssä.

Varmuuskopiot tallennetaan tiedosto-muotoiseen storageen tiedostomuodossa. Tiedostossa voidaan käyttää LZO- tai GZIP-pakkausta tilan säästämiseksi. Snapshotteja ei sisällytetä varmuuskopioihin.

Varmuuskopioon voidaan palata ylikirjoittamalla virtuaalikone varmuuskopion datalla.



#### 4.10 High Availability

Jotta klusterissa ei tapahtuisi virhetilanteissa tiedon korruptoitumista tai muuta ikävää, tarvitaan toimiva fencing. Fencingillä tarkoitetaan jonkinlaista skriptattavaa komento-rajapintaa, jolla fyysisestä hostista voidaan katkaista virta. Tällaisia ovat esimerkiksi Dellin palvelimista löytyvä iDRAC ja APC:n UPS:it sekä geneerinen IPMI.

Toimiva ja testattu fencing on erittäin tärkeä osa High Availability klusteria, joten se täytyy luoda ensin. Fencing-toiminto täytyy ottaa käyttöön kaikilla hosteilla muokkaamalla ”/etc/default/redhat-cluster-pve”-tiedostoa.

```
FENCE_JOIN="yes"
```

Muutoksen jälkeen voidaan liittyä fence-domainiin ja tarkastaa sen status. Komennot täytyy ajaa kaikilla hosteilla.

```
fence_tool join
fence_tool ls
```

Täytyy muokata /etc/pve/cluster.conf -tiedostoa ja tehdä sinne fencedevide merkinnät. Nämä ovat hyvin erilaiset riippuen millaista fencing-tapaa käytetään. Tapoja on esimerkiksi IPMI tai Dell DRAC5. Seuraavassa esimerkissä käytämme IPMI:ä, koska käytössä oli Dellin palvelimia, joissa on buginen iDRAC6.

IPMI vaatii toimiakseen ipmitool-paketin jokaisella hostilla. Sen voi asentaa komentoriviltä.

```
aptitude install ipmitool
```

Jos hosteja on vain kaksi kappaletta, täytyy <cman> -tagien sisään lisätä kaksi parametria: two\_node="1" ja expected\_votes="1".

Aina kun tiedostoa muokataan, täytyy muistaa kasvattaa ”config\_version”-parametria yhdellä.

```
<?xml version="1.0"?>
<cluster config_version="4" name="proxmox-cluster">
  <cman expected_votes="1" keyfile="/var/lib/pve-
cluster/corosync.authkey" two_node="1"/>
  <clusternodes>
    <clusternode name="kvm1" nodeid="1" votes="1">
      <fence>
        <method name="1">
```

```

        <device name="ipmi1"/>
    </method>
</fence>
</clusternode>
<clusternode name="kvm3" nodeid="2" votes="1">
    <fence>
        <method name="1">
            <device name="ipmi3"/>
        </method>
    </fence>
</clusternode>
</clusternodes>
<fencedevices>
    <fencedevice agent="fence_ipmilan" name="ipmi1"
lanplus="1" ipaddr="192.168.200.251" login="root" passwd="xxxxxx"
power_wait="5"/>
    <fencedevice agent="fence_ipmilan" name="ipmi3"
lanplus="1" ipaddr="192.168.200.253" login="root" passwd="xxxxxx"
power_wait="5"/>
</fencedevices>
</cluster>

```

Klusterin konfiguraation xml-formaatti on hyvin tarkka ja monimutkainen. Siitä löytyy lisätietoa komennolla ”man cluster.conf”.

Muokkauksen jälkeen voidaan mennä webui:ssa tarkistamaan konfiguraation. HA-välilehti löytyy Proxmox VE:n käyttöliittymässä Datacenter-kohdan alla.

Jos välilehdelle mennessä tulee error-viesti tyyliin: “mismatched tag at line 21, column 2, byte 524 at /usr/lib/perl5/XML/Parser.pm line 187 (500)”, kyseessä on kirjoitusvirhe äsken tehdyssä konfiguraatiossa. Konfiguraation voi myös validoida komentoriviltä.

```
ccs_config_validate -v
```

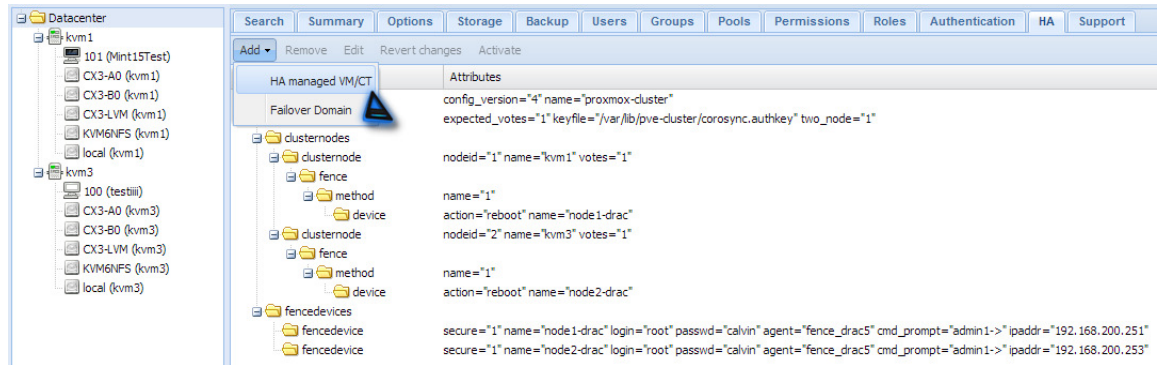
Jos konfiguraatiotiedosto on virheetön, sivun alalaidassa pitäisi näkyä muutokset ja Activate-nappi olla painettavissa. Painamalla sitä muutokset astuvat voimaan.

Fencingin toimivuutta voi testata esim. katkaisemalla kylmästi virrat yhdestä hostista tai komennolla komentoriviltä.

```
fence_node kvm1 -vv
```

Tapahtumia voi seurata hengissä olevan hostin Syslog-välilehdeltä.

Jos fencing toimii ongelmitta, voidaan virtuaalikoneita lisätä HA-konfiguraatioon. Virtuaalikoneesta täytyy tietää sen uniikki ID-numero. Virtuaalikoneen täytyy olla sammutettuna. Muutokset täytyy aktivoida ennen kuin ne tulevat voimaan.



Kuva 20. HA-näkymä Proxmox VE:ssä.

HA:n asettamisen jälkeen sitä voi testata raa’asti katkaisemalla virrat hostista. Testeissä virtuaalikone nousi pystyyn uudella hostilla noin puolessa minuutissa.

## 5 POHDINTA

Hypervisorina KVM tuntuu olevan erittäin pätevä. Sen avoimen lähdekoodin tuoma monimuotoisuus ja joustavuus ovat vahvoja etuja. Itsessään KVM ei tarjoa paljoa muuta kuin pelkän hypervisorin, mutta muiden ohjelmistojen, kuten libvirt:in, VirtIO:n ja Open vSwitch:in, kautta sen ominaisuudet vastaavat kaupallisia tuotteita. Tästä kolmannen osapuolen ohjelmistoihin ja rajapintoihin tukeutumisesta tulee niin positiivisia kuin negatiivisia vaikutuksia. Positiivisena voidaan pitää kehityksen hajautuneisuutta. Eri ominaisuuksista vastaavat erillään toimivat ryhmät ja ne pystyvät kehittämään oman sovelluksensa ominaisuuksia vapaasti. Tästä myös syntyy negatiivinen puoli, sillä jonkin ns. korkeammalla toimivan ohjelmiston kehittäjät eivät voi toteuttaa ominaisuuksia ennen kuin ne ovat tuettuna ns. alemmalla toimivassa rajapinnassa.

Käytettävyys riippuu siitä, mitä käyttöliittymää käytetään. Moni tarjolla olevista käyttöliittymistä on ns. pilvipalvelun alusta, jossa on virtuaalikoneita lähestytty valmiiksi rakennettujen templateiden kautta ja ovat muutenkin selvästi tarkoitettu palveluntarjoajille. Virtualisointialustan hallintaan tarkoitettuja sovelluksia löytyy reippaasti vähemmän ja Proxmox VE tuntuu ainakin allekirjoittaneen mielestä ainoalta vakavasti otettavalta vaihtoehdolta.

Ylläpidollisesti KVM on selvästi astetta haastavampi kuin esimerkiksi VMwaren tuotteiden käyttö, varsinkin jos halutaan käyttää sitä täysin ilman graafisia käyttöliittymiä. Vaikka osassa KVM:n ympärille rakennetuista ratkaisuista suurin osa päivittäisistä toiminnoista voidaankin helposti suorittaa graafisen käyttöliittymän avulla, joudutaan niissäkin usein palaamaan tuttuun Linux-komentoriviin suorittamaan komentoja. Jos lähdetään siltä pohjalta, että käytetään KVM virtualisointia maksutta, kaikki tuki ja dokumentaatio on erinäisten Internet forumien ja viime kädessä Googlen hakupalvelun varassa. Vaikka KVM:stä onkin jo paljon tietoa ympäri Internetiä, voi joihinkin ongelmiin vastaukset olla hyvinkin suuren ns. kiven takana.

Yhtenäisen käyttöliittymän puuttuminen on suuri puute, joka voi vähentää kiinnostusta KVM:ää kohtaan. Itselläni ei opinnäytteen aikana tullut juuri kokemusta muista käyttöliittymistä kuin Proxmox VE:stä ja komentoriviltä ajettavista komennoista, mutta Tarmo Kääriäisen samaan aikaan tekemästä opinnäytteestä, joka käsittelee KVM:n hallintatyökaluja, käy

ilmi etteivät muutkaan käyttöliittymät ole vielä sillä tasolla, että komentoriviin ei tarvitsisi turvautua.

Ymmärrettävistä syistä pelkästään komentoriviltä käytettävää KVM:ää ei voi suositella, jos ylläpitäjiä tai hallintatoimintojen käyttäjiä on useita. Yksin tai pienellä porukalla hoidettuna asioista pysynee perillä vielä jotenkuten, mutta suuremmalla käyttäjämäärällä käyttöoikeuksien ja toimintojen hallinta muodostuu liian hankalaksi. Koska Proxmox VE:n kaltaisia sovelluksia on olemassa, en näekään KVM:ää komentoriviltä käytettynä muuhun kuin testi- ja oppimiskäyttöön sopivaksi. Komentorivin komentoja on kuitenkin KVM-ylläpitäjän hyvä osata, sillä monissa käyttöliittymissä ongelmanratkointia ja esimerkiksi klusterin hallinta tapahtuu komentoriviltä.

Proxmox VE vaikutti testien perusteella varsin pätevältä ratkaisulta. Sen asennusprosessi on itsensä selittävä ja se asentaa käytännössä kaiken tarpeellisen. Web-käyttöliittymä on selkeä, eikä komentoriviin tarvinnut virtuaalikoneiden hallinnassa koskea. Myös verkkojen ja storage-laitteiden hallinta onnistuu web-käyttöliittymästä suurimmilta osin.

Siirtyminen KVM:ään jostain muusta hypervisorista vaatii virtuaalikoneiden asetusten tarkasteluja ja uudelleenluontia uuteen ympäristöön sekä levyjen konvertoinnin KVM:n tukemaan formaattiin. Yleisesti esimerkiksi VMware-ympäristössä on käytössä VMFS-storaget, joihin tallennetaan virtuaalikoneiden levyt tiedostomuodossa, mutta KVM:n yleinen käytäntö on qcow2-formaatin tiedostomuotoiset levyt tai vaihtoehtoisesti LVM-osiot. Varsinkin Windows-käyttöjärjestelmää ajavissa virtuaalikoneissa voi tulla vastaan myös ajuriongelmia, jos virtuaalilaitteissa on erikoisuuksia.

Syytä siirtyä jostain muusta hypervisorista KVM:n käyttöön ei tarvitse kaukaa hakea. Ilmaisuus on vahva valttikortti varsinkin pienemmissä ympäristöissä, joissa virtualisoinnin lisenssimaksut voivat olla suuri osa budjetista. Suuremmissa ympäristöissä, joissa hallitaan suuria määriä virtuaalikoneita, hosteja ja käyttäjiä, voi KVM olla vielä hieman raakile. Tokihan hyviä hallintajärjestelmiä jo KVM:n ympärille löytyy, mutta ne ovat lähinnä pilvipalveluiden pyörittäjille tarkoitettuja.

Testauksissa lähdin liikkeelle aluksi täysin oman tietotaitoni pohjalta nähdäkseni kuinka pitkälle sillä pääsee. Proxmox VE:n itsensä selittävä asennus ja yksinkertainen käyttöliittymä veivätkin pitkälle kunnes klusterin teossa oli pakko ajaa komentoja komentoriviltä, jolloin jouduin hakemaan tietoa Proxmoxin viralliselta wiki-sivustolta. Täysin komentoriviltä ajetta-

van KVM:n kanssa jouduin käyttämään tiedonlähteenä erilaisia foorumeja, blogeja ja wiki-sivustoja lähes koko ajan.

Olen itse opinnäytteen aikana siirtynyt omassa kotona pyörittämässäni yhden palvelimen virtualisointiympäristössäni Proxmox VE:hen. Aiemmin pyöritin KVM:ää Fedora 14 -distribuution päällä ja käytin virtuaalikoneita virt-manager ohjelmalla. Proxmox VE:n kanssa ei ole ollut mitään ongelmia ja se tuntuu olevan hyvin vakaa. Käytettävyys pienessä ympäristössä on ollut hyvinkin riittävä.

KVM:n tulevaisuus näyttää valoisalta ja se tuntuu kehittyvän nopeasti. Erilaisia tekniikoita tulee itse hypervisorin rinnalle täydentämään puuttuvia ominaisuuksia ja niitä integroidaan käyttöliittymiin. Proxmox VE on hyvä esimerkki, sillä opinnäytteen aikana 3.2-version myötä siihen lisättiin Open vSwitch -tuki, joka laajentaa virtuaalista verkotusta Proxmox VE:n sisällä huomattavasti. Muita suurempiin ympäristöihin tarpeellisia ominaisuuksia, kuten load-balancingia saadaan vielä odottaa, mutta ennustan niiden tuloa KVM:n maailmaan helpommin saataviksi lähivuosien aikana.

## LÄHTEET

eWeek. How Did KVM Virtualization Get Into the Linux Kernel? 2014. Saatavilla: <http://www.eweek.com/cloud/how-did-kvm-virtualization-get-into-the-linux-kernel.html> (Luettu 7.4.2014)

IBM a. Why KVM? - KVM in action. 2014. Saatavilla: <http://www-03.ibm.com/systems/virtualization/kvm/whykvm.html> (Luettu 17.4.2014)

IBM b. Best practice: Use block devices for VM storage. 2014. Saatavilla: <http://pic.dhe.ibm.com/infocenter/lxinfo/v3r0m0/index.jsp?topic=%2Fliaat%2Fliaatbpblock.htm> (Luettu 7.4.2014)

KVM a. FAQ. 2014. Saatavilla: <http://www.linux-kvm.org/page/FAQ> (Luettu 7.4.2014)

KVM b. Guest Support Status Saatavilla: [http://www.linux-kvm.org/page/Guest\\_Support\\_Status](http://www.linux-kvm.org/page/Guest_Support_Status) (Luettu 7.4.2014)

KVM c. Setting guest network. 2014. Saatavilla: <http://www.linux-kvm.org/page/Networking> (Luettu 7.4.2014)

Kääriäinen, T. 2014. KVM-Hallintatyökalut, opinnäytetyö. Kajaani.

Open vSwitch. Documentation › Configuration Cookbook › VLANs. 2014. Saatavilla: <http://openvswitch.org/support/config-cookbooks/vlan-configuration-cookbook/> (Luettu 17.4.2014)

Proxmox a. About Proxmox. 2014. Saatavilla: <http://www.proxmox.com/about> (Luettu 7.4.2014)

Proxmox b. Documentation - Storage Model. 2014. Saatavilla: [https://pve.proxmox.com/wiki/Storage\\_Model](https://pve.proxmox.com/wiki/Storage_Model) (Luettu 7.4.2014)

Proxmox c. System Requirements. 2014. Saatavilla: <http://www.proxmox.com/proxmox-ve/requirements> (Luettu 7.4.2014)

Proxmox d. Roadmap. 2014. Saatavilla: <https://pve.proxmox.com/wiki/Roadmap> (Luettu 7.4.2014)

Proxmox e. Subscription Plans. 2014. Saatavilla: <http://proxmox.com/proxmox-ve/pricing> (Luettu 25.3.2014)

Ubuntu. Ubuntu documentation - UbuntuBonding. 2014. Saatavilla: <https://help.ubuntu.com/community/UbuntuBonding> (Luettu 7.4.2014)

VMware. Minimum system requirements for installing ESXi/ESX. 2014. Saatavilla: [http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayK&externalId=1003661](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayK&externalId=1003661) (Luettu 7.4.2014)

Vyom. Libvirt - The Unsung Hero of Cloud Computing. 2013. Saatavilla: [http://www.vyomtech.com/2013/12/17/libvirt\\_the\\_unsung\\_hero\\_of\\_cloud\\_computing.html](http://www.vyomtech.com/2013/12/17/libvirt_the_unsung_hero_of_cloud_computing.html) (Luettu 7.4.2014)

Wikipedia a. Kernel-based Virtual Machine. 2014. Saatavilla: [http://en.wikipedia.org/wiki/Kernel-based\\_Virtual\\_Machine](http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine) (Luettu 7.4.2014)